

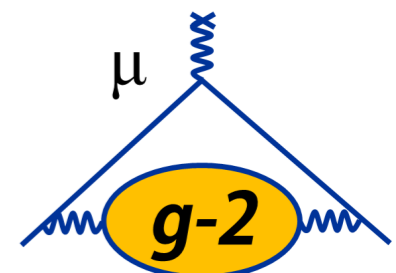


Tip-to-tail SLAC: Calorimeter test run

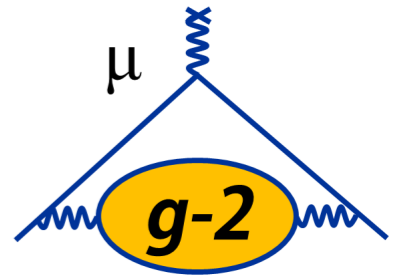
Kim Siang Khaw

Muon $g-2$ Computing Review

Nov 07-08, 2016

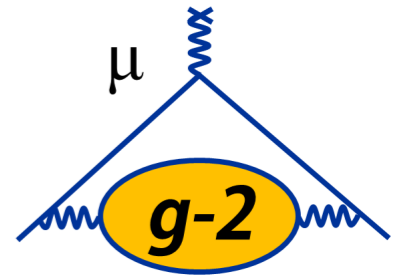


Content



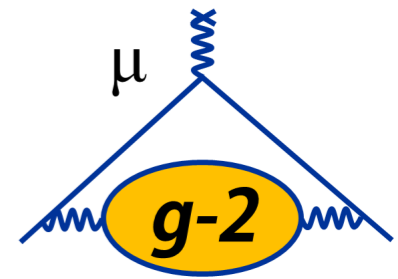
- Motivation
- Overview of SLAC calorimeter test run
- Offline computing framework
- User friendly tools
- Summary

Motivation



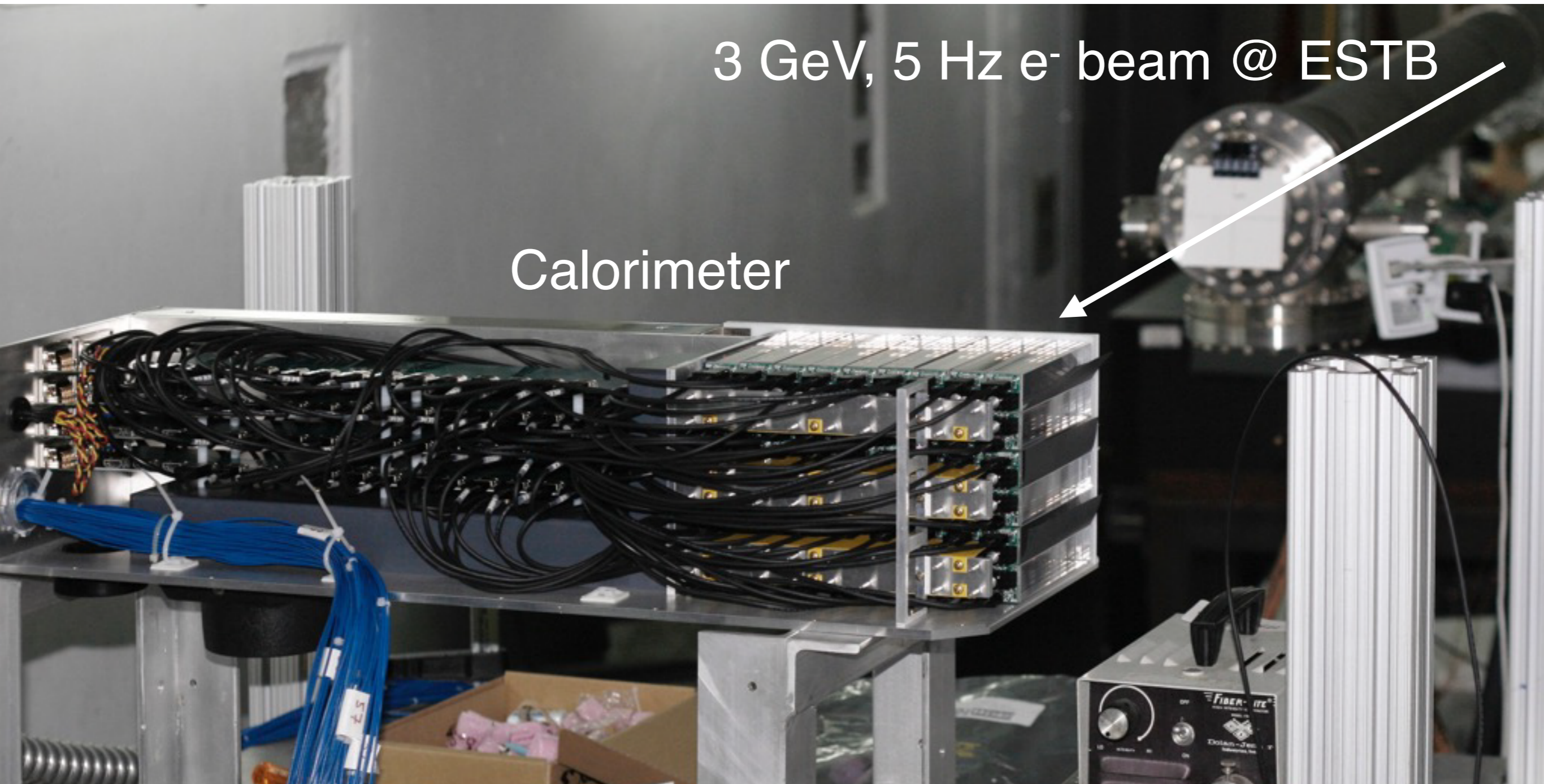
- End-to-end test of the calorimetry system
 - Lead Fluoride (PbF_2) Cherenkov calorimeter
 - Laser-based calibration and monitoring system
 - Custom 800 MSPS fast digitizer
 - FC7 trigger and clock distribution system
 - DAQ (frontend and backend)
 - ***art*-based offline computing framework**
- Characterization of the calorimetry system
 - Energy and timing resolution of the calorimeter
 - Stability of the energy scale
 - etc

SLAC End Station A

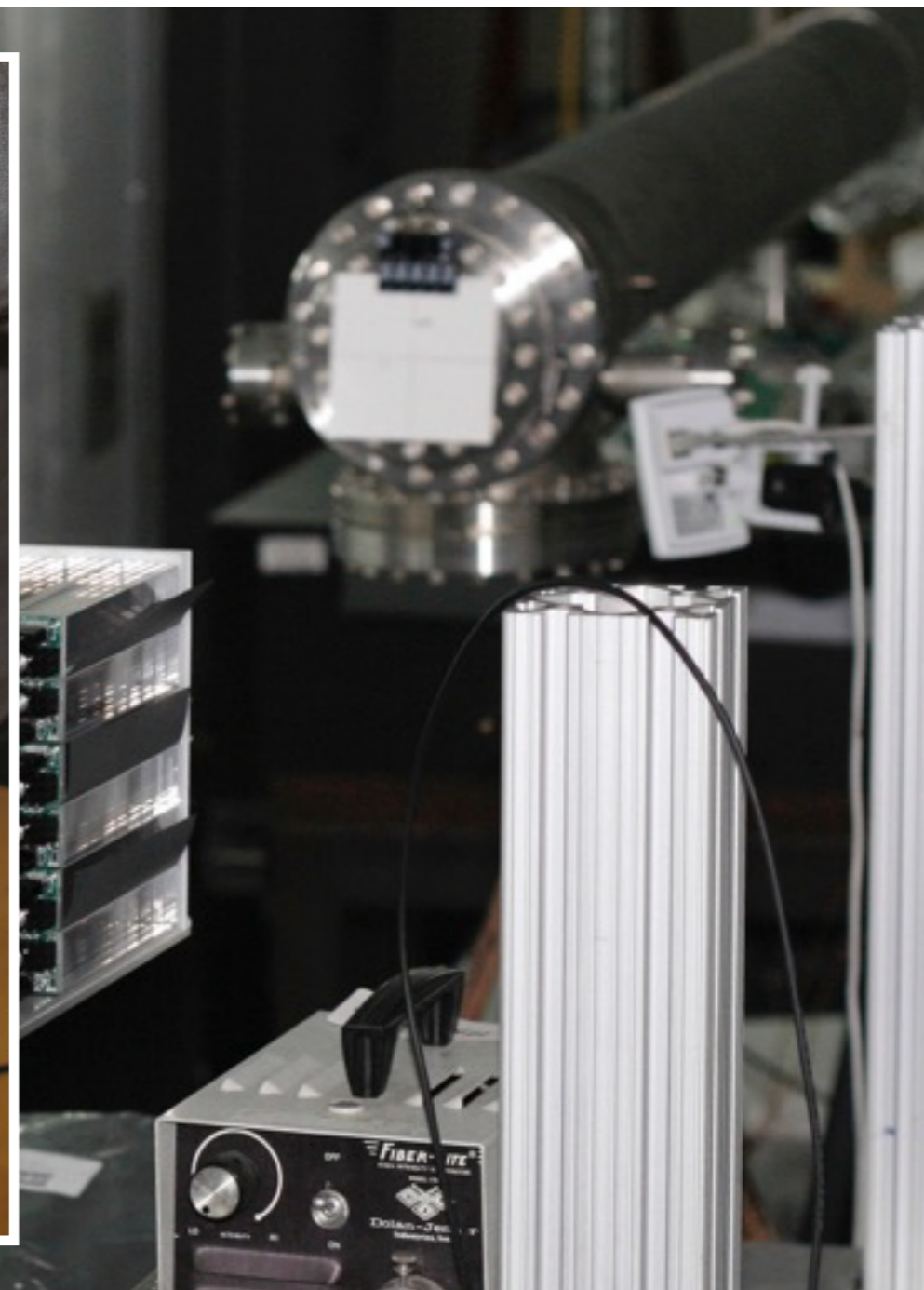
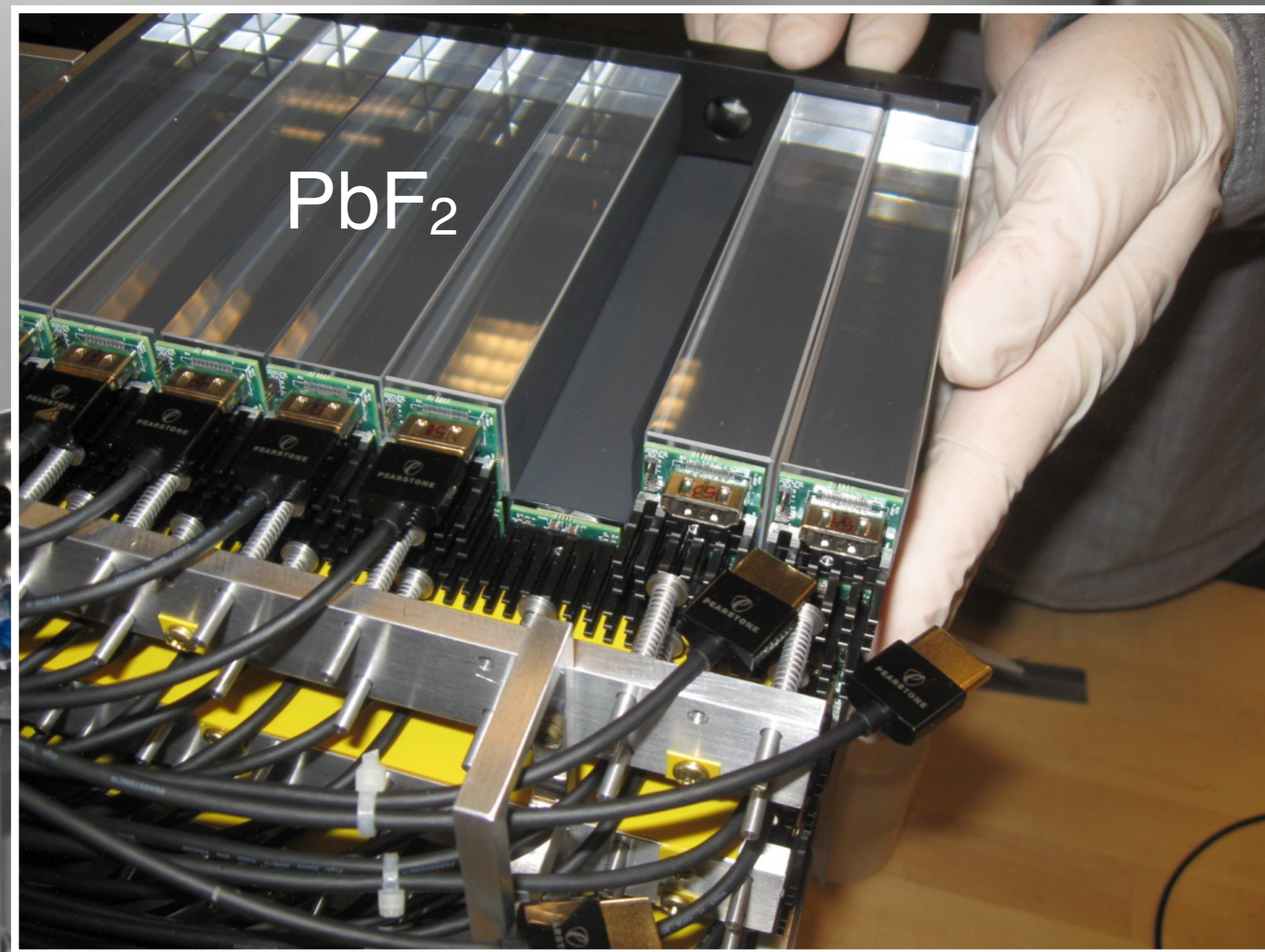
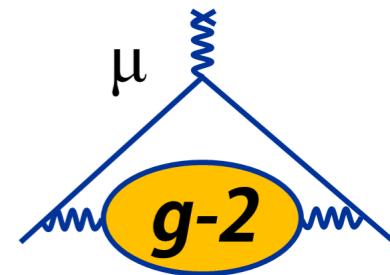


3 GeV, 5 Hz e^- beam @ ESTB

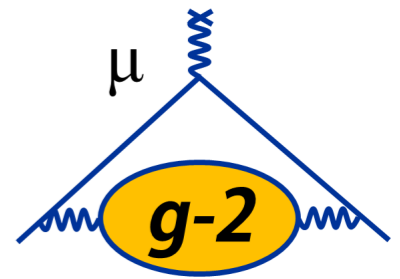
Calorimeter



Calorimeter (PbF₂ + SiPM)

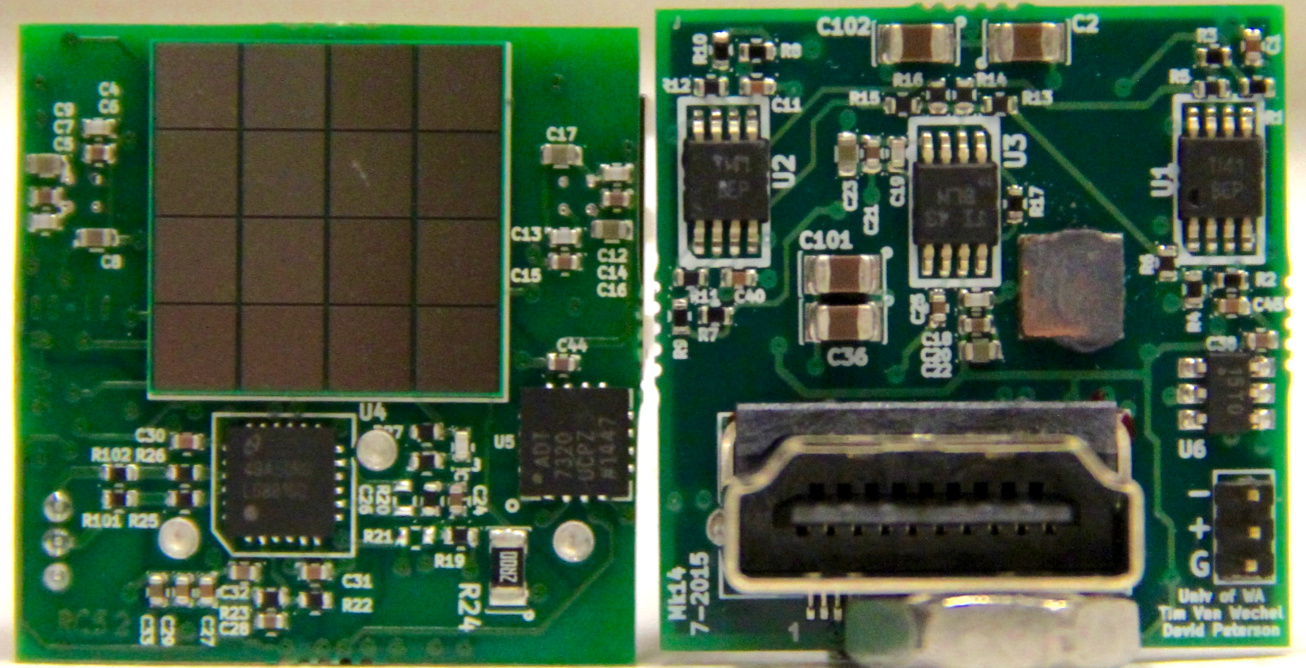


Calorimeter (PbF₂ + SiPM)

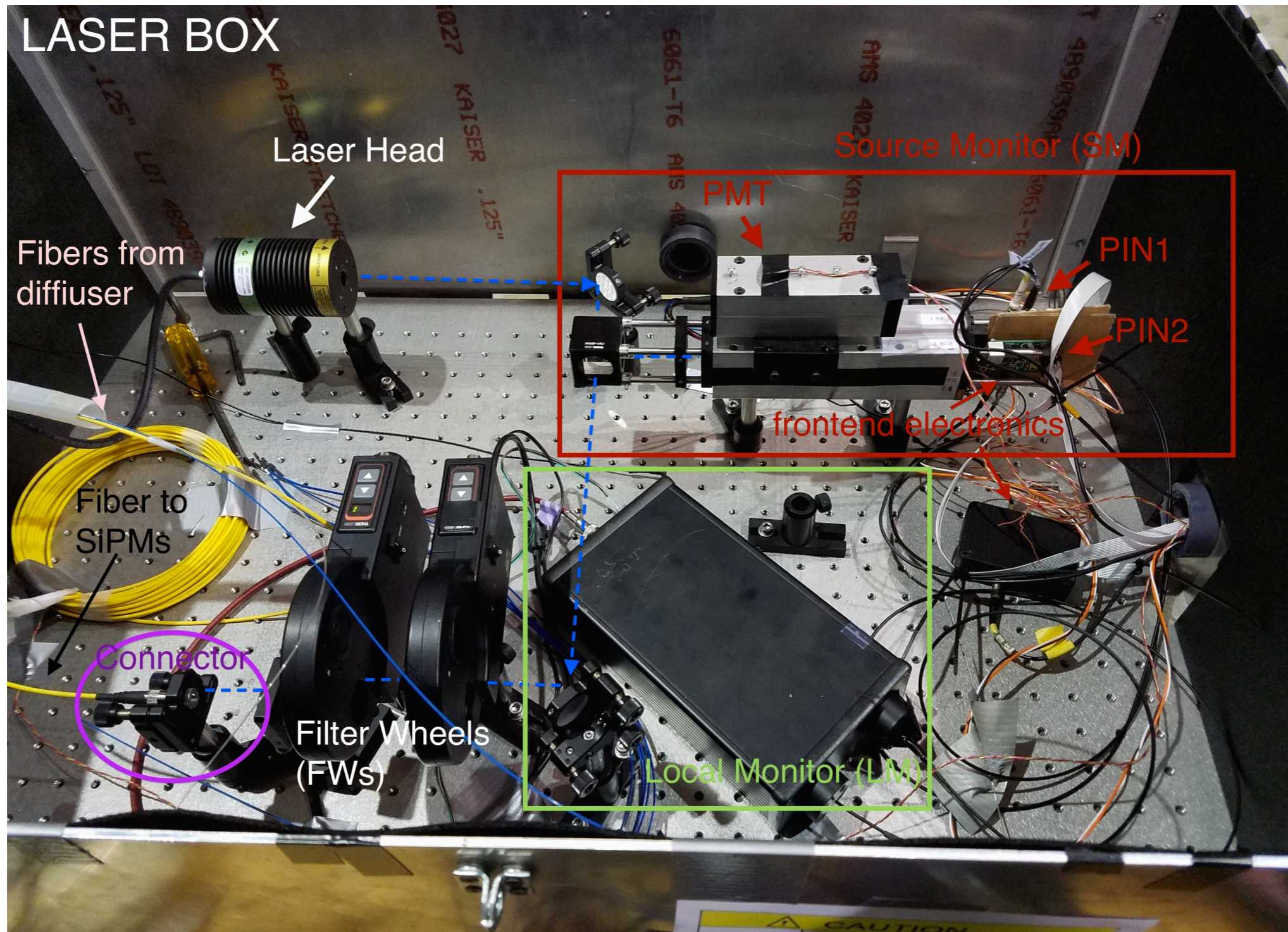
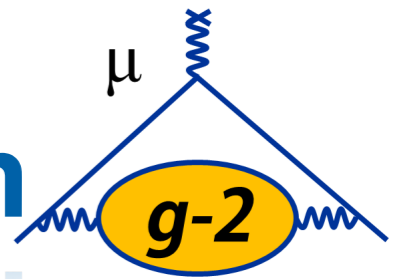


PbF₂

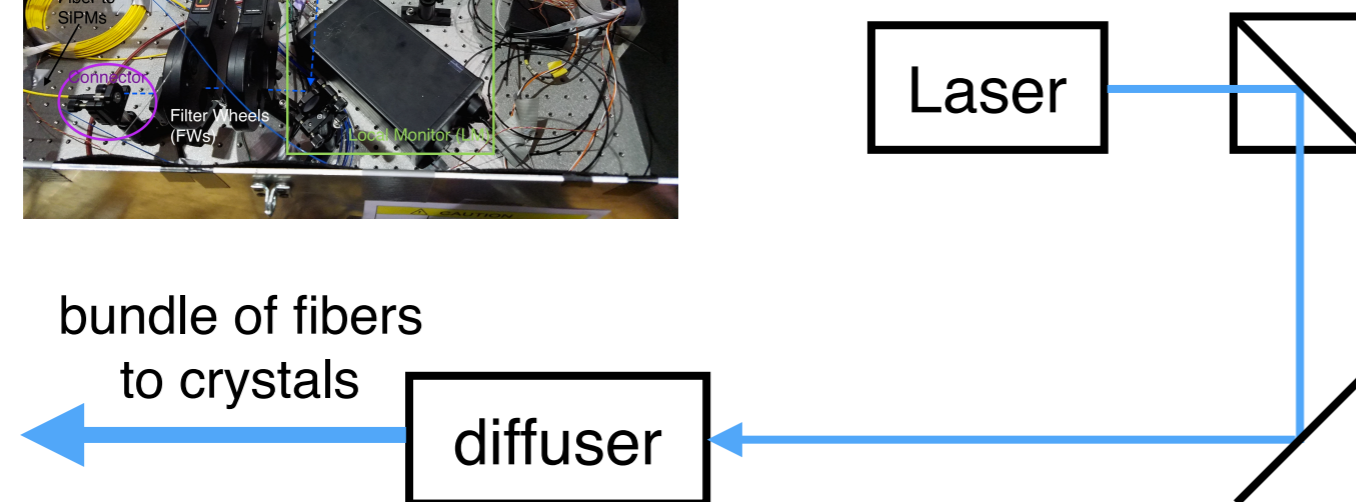
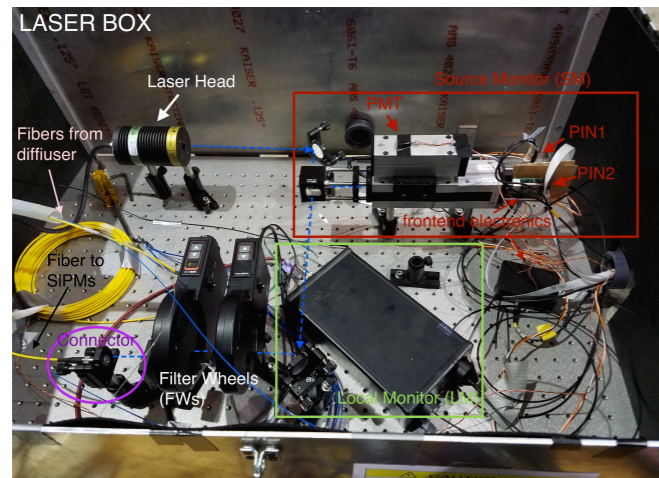
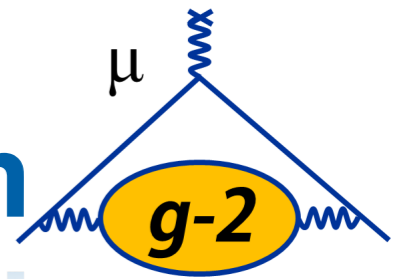
SiPM pre-amp board



Laser calibration and monitoring system

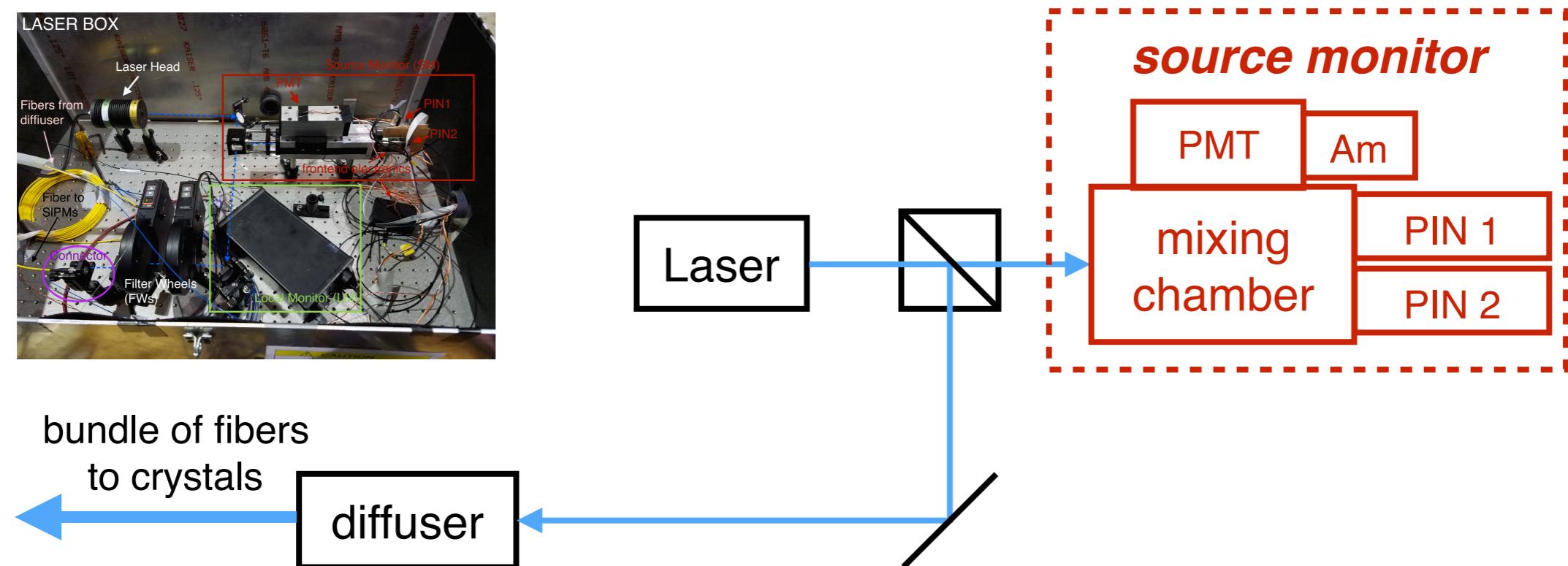
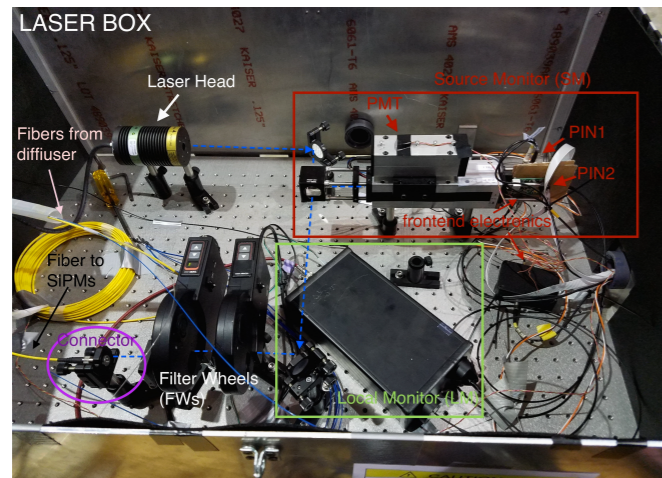
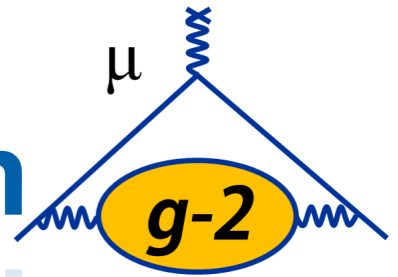


Laser calibration and monitoring system



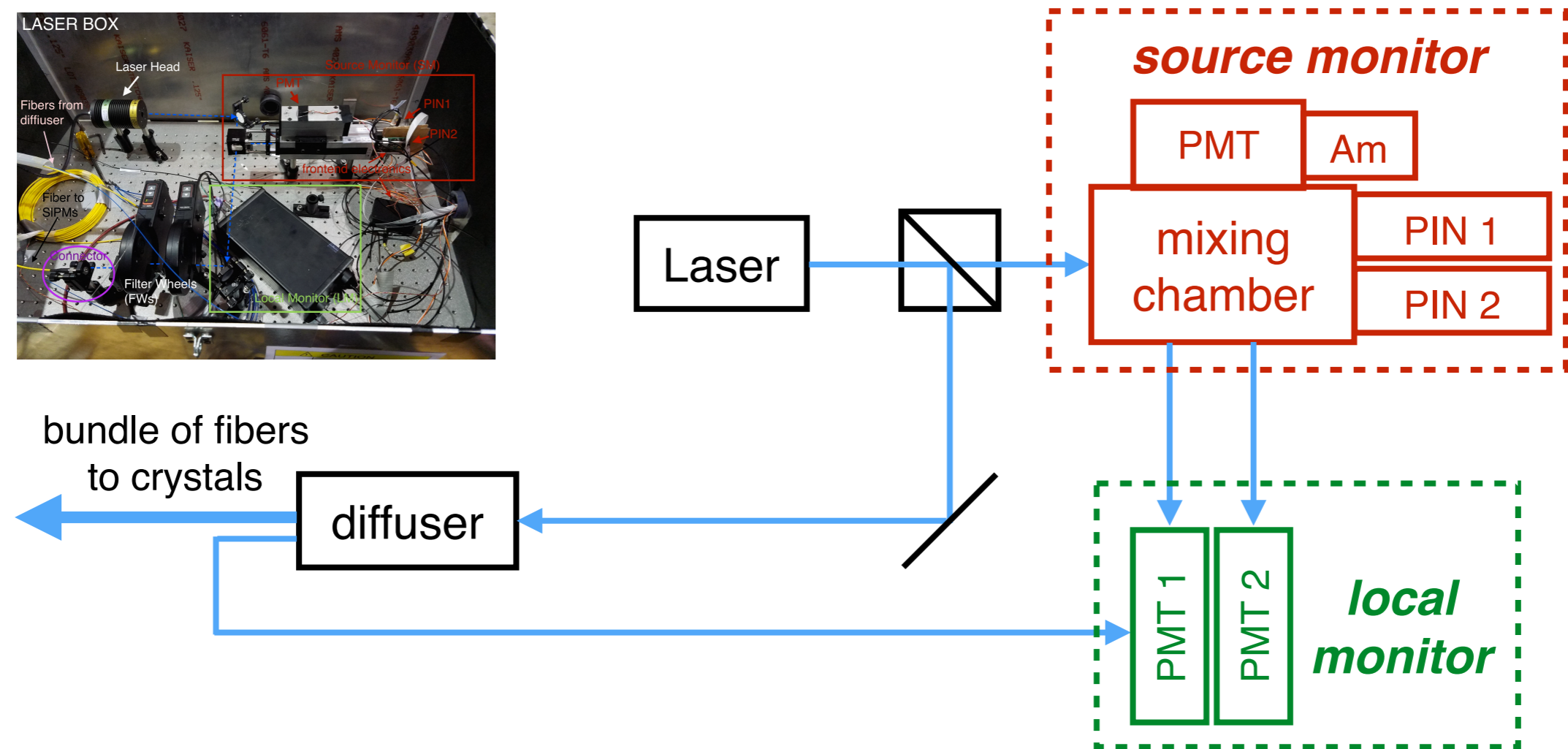
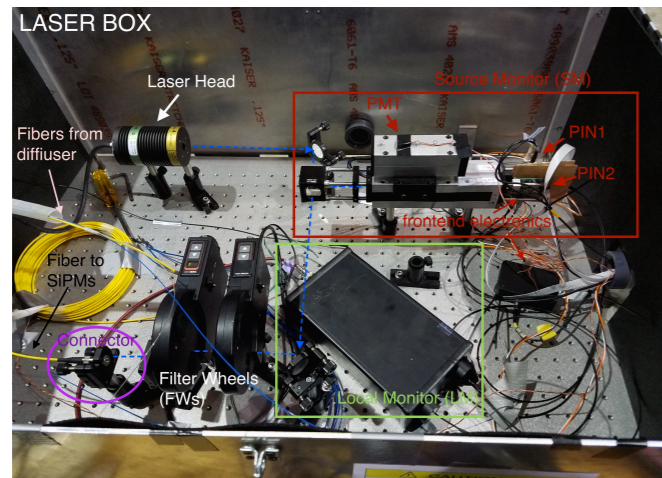
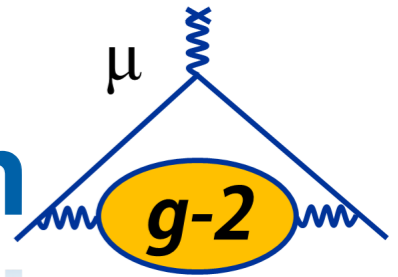
- SiPM gain monitoring - high stability 405 nm PicoQuant laser

Laser calibration and monitoring system



- SiPM gain monitoring - high stability 405 nm PicoQuant laser
- **Laser intensity monitoring - Source monitor (PINs & PMT)**

Laser calibration and monitoring system



- SiPM gain monitoring - high stability 405 nm PicoQuant laser
- **Laser intensity monitoring - Source monitor (PINs & PMT)**
- **Laser light distribution chain monitoring - Local monitor (PMTs)**
- Target SiPM gain stability $\sim 10^{-4}$ /hour

End-to-end system

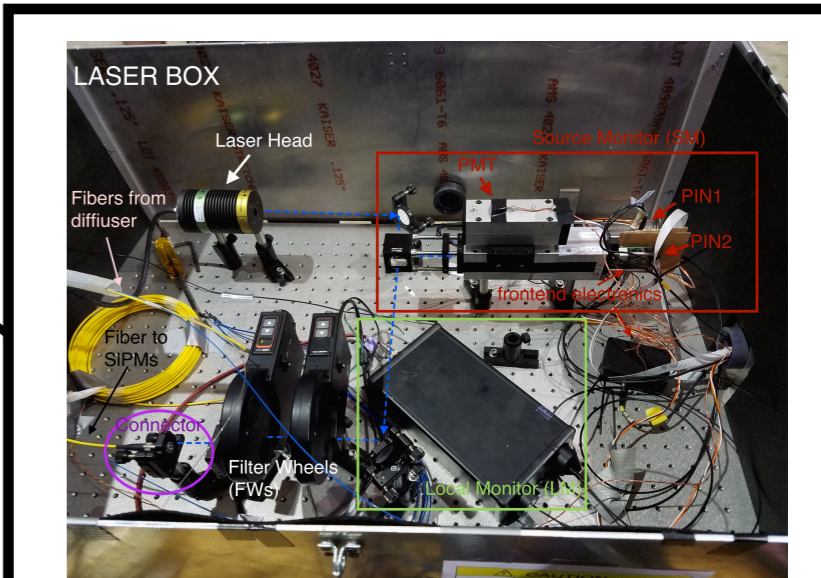
Signal digitization

Trigger clock
distribution
system (TCDS)



calorimeter

Frontend:
mCPU + GPU
processing



laser calibration and monitor

online
data quality
monitor

Run Status				
<div>Run 2149</div> <div>Stopped</div> <div>Start</div>	Start: Sun Jun 5 16:58:00 2016		Stop: Sun Jun 5 17:04:22 2016	
	Alarms: On	Restart: No	Data dir: /data/slac/	
	Experiment Name: SLAC			
	Rider status: 0			
	17:04:33 [Logger,INFO] Run #2149 stopped			
Equipment				
Equipment	Status	Events	Events[/s]	Data[MB/s]
EB	Ebuilder@g2be	1880	0.0	0.000
MasterGM2	MasterGM2@g2be	0	0.0	0.000
AMC1301	AMC1301@g2calo	1880	0.0	0.000
AMC1302	AMC1302@g2calo	1880	0.0	0.000
AMC1303	AMC1303@g2calo	1880	0.0	0.000
Temperature	Ok	28	0.9	0.000

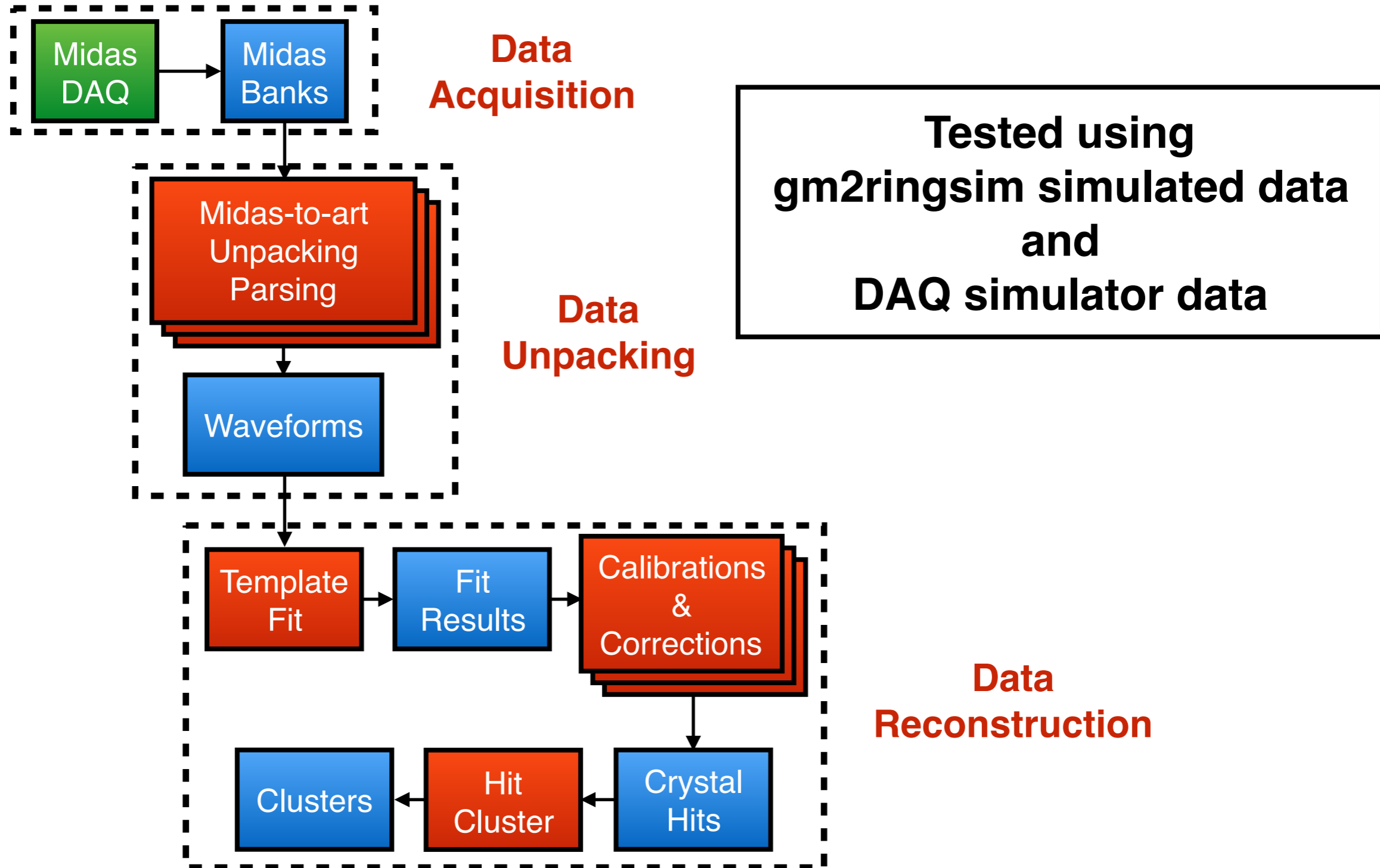
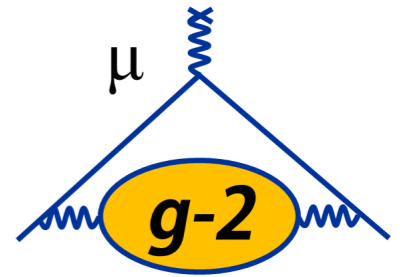
Data
storage

analysis
machine

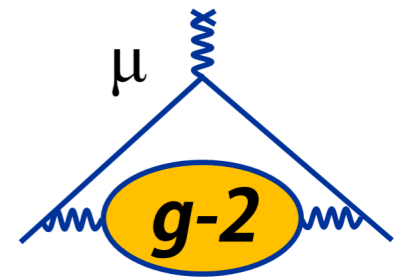
offline processing

Backend: MIDAS event builder

Offline framework (Pre-SLAC)



MIDAS-to-art



MIDAS Event Structure

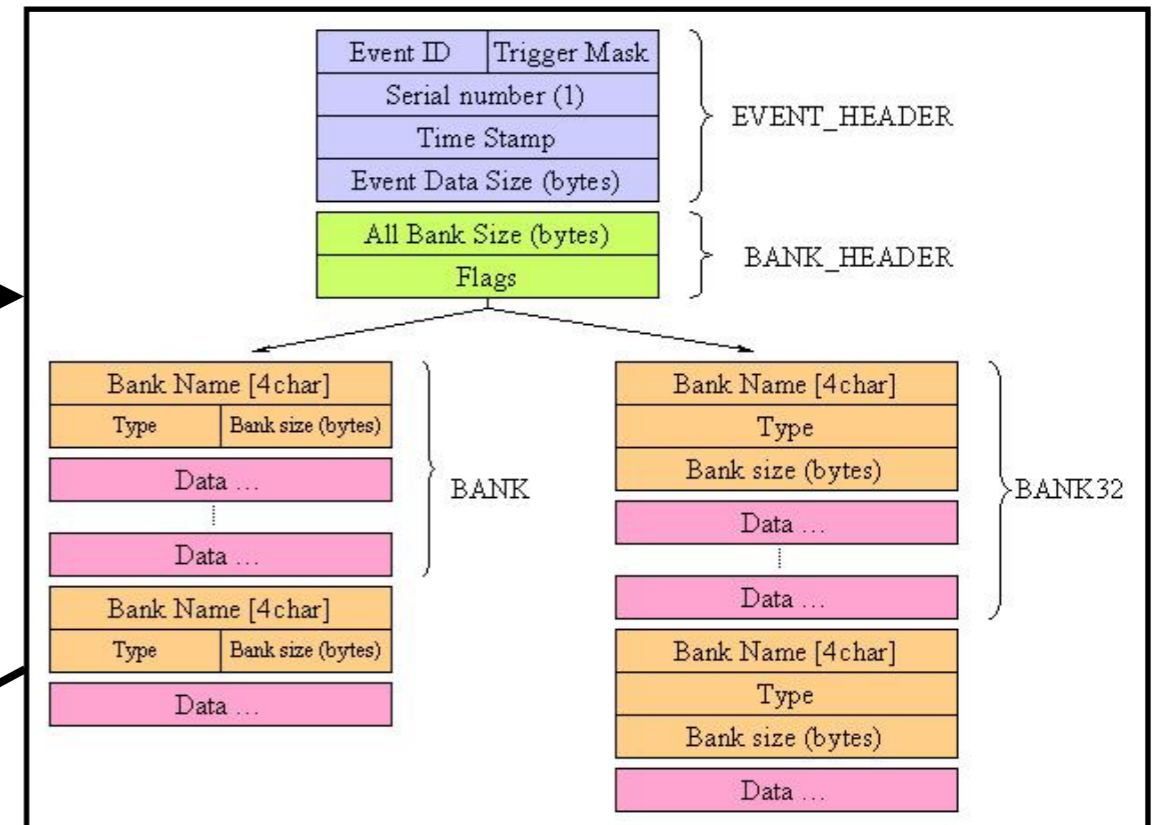
Run Status

Run 2149 Stopped
Start: Sun Jun 5 16:58:00 2016 Stop: Sun Jun 5 17:04:22 2016
Alarms: On Restart: No Data dir: /data/slac/
Experiment Name: SLAC
Rider status: 0
17:04:33 [Logger,INFO] Run #2149 stopped

Equipment

Equipment	Status	Events	Events[/s]	Data[MB/s]
EB	Ebuilder@g2be	1880	0.0	0.000
MasterGM2	MasterGM2@g2be	0	0.0	0.000
AMC1301	AMC1301@g2calo	1880	0.0	0.000
AMC1302	AMC1302@g2calo	1880	0.0	0.000
AMC1303	AMC1303@g2calo	1880	0.0	0.000
Temperature	Ok	28	0.9	0.000

output
MIDAS file



Bank:CT01 Length: 322204(I*1)/80551(I*4)/161102(Type)

```

1-> 30024 2 71 54 196 0 14149 0
9-> 42 0 1801 1805 1799 1801 1794 1807
17-> 1804 1802 1794 1775 1158 -119 -599 -39
25-> 933 1609 1805 1767 1705 1719 1769 1819
33-> 1858 1879 1894 1900 1882 1874 1851 1850
41-> 1840 1844 1847 1869 1886 1905 1902 1899
49-> 1880 1864 1832 1820 1746 1763 1756 1755
57-> 1741 1754 1753 1603 620 -284 -239 514
    
```

dumped data

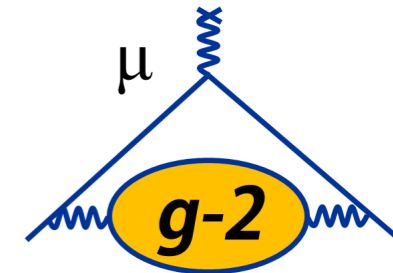
MIDAS-to-art

```

gm2midastoart::CaloRawBankgm2midastoart::MidasEventArtRecords_
gm2midastoart::CaloTBankgm2midastoart::MidasEventArtRecords_Mi
gm2midastoart::CaloTBankgm2midastoart::MidasEventArtRecords_Mi
    
```

stored as data products (header, data) and put into art event

Data unpacking and parsing



Bank: **CT01** Length: 322204(I*1)/80551(I*4)/161102(Type)

1->	30024	2	71	54	196	0	14149	0
9->	42	0	1801	1805	1799	1801	1794	1807
17->	1804	1802	1794	1775	1158	-119	-599	-39
25->	933	1609	1805	1767	1705	1719	1769	1819
33->	1858	1879	1894	1900	1882	1874	1851	1850
41->	1840	1844	1847	1869	1886	1905	1902	1899
49->	1880	1864	1832	1820	1746	1763	1756	1755
57->	1741	1754	1753	1603	620	-284	-239	514

Simple data format
(GPU pre-processed
data, DAQ timing data)

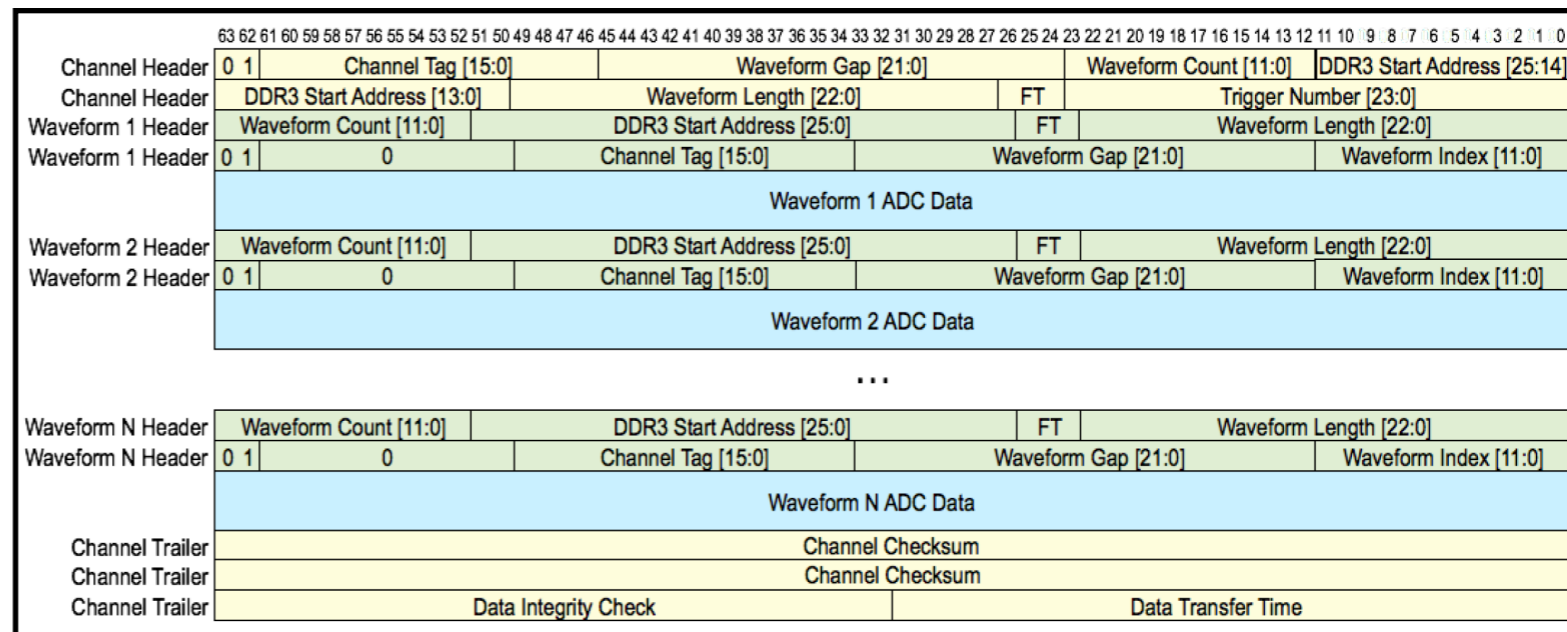


art data products

Waveforms (chopped)

Headers

Waveforms (raw)



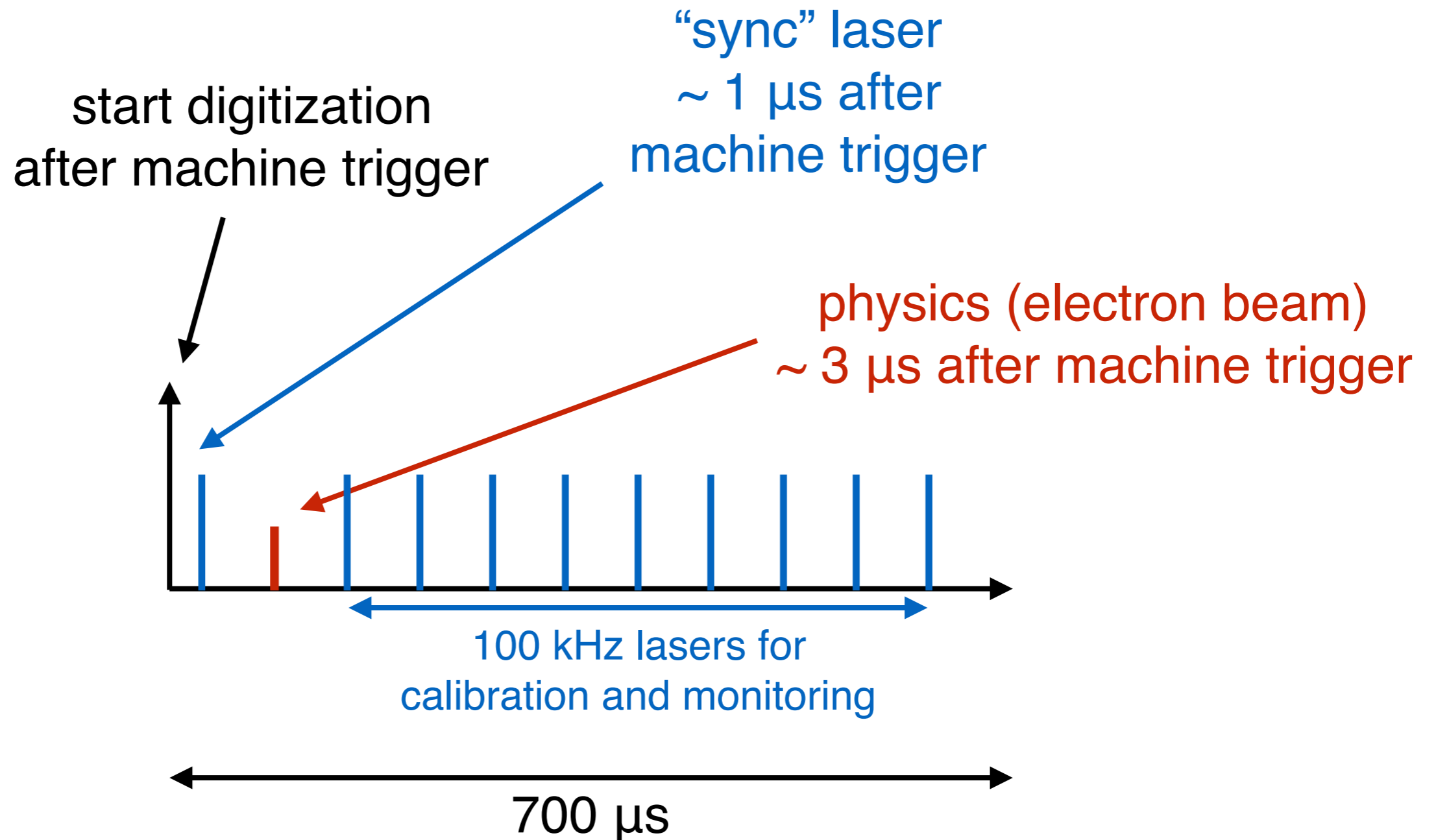
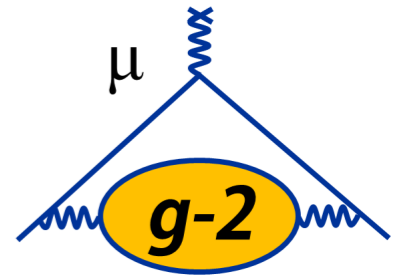
Raw digitizer data

gm2parser classes

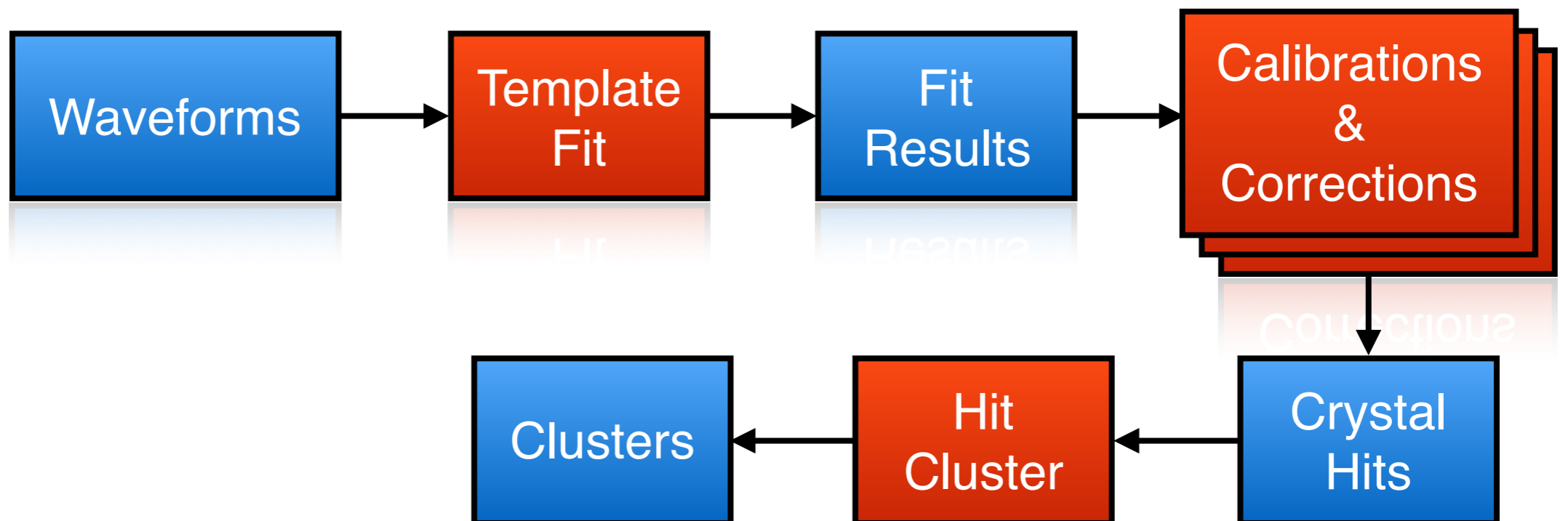
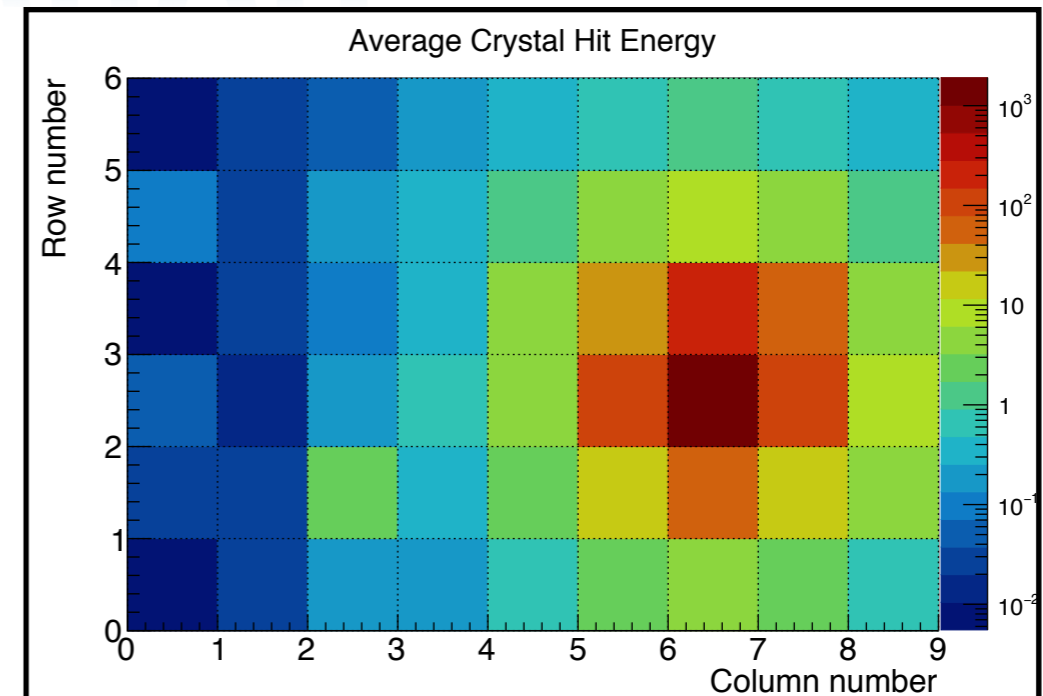
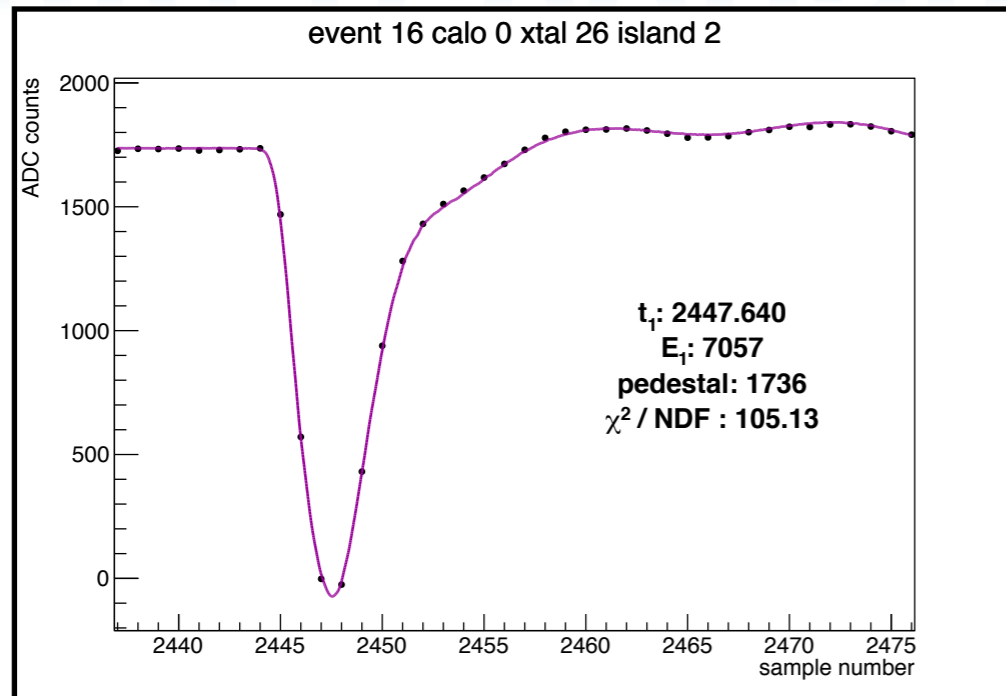
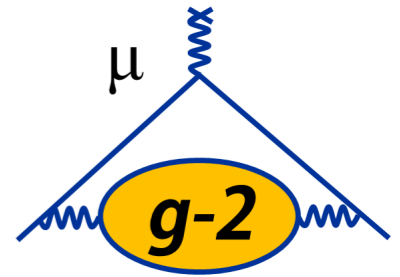
```
gm2parser::FedRider fedRiderCH;
fedRiderCH.AddRider(channelData);

if(verboseLevel_ > 1){
    std::cout<<"\t\t----> Entering Channel #"<<fedRiderCH.ChannelTag()<<std::endl;
    std::cout<<"\t\tFillType: "<<fedRiderCH.FillType()<<std::endl;
    std::cout<<"\t\tWaveformGap: "<<fedRiderCH.WaveformGap()<<std::endl;
    std::cout<<"\t\tWaveformCount: "<<fedRiderCH.WaveformCount()<<std::endl;
    std::cout<<"\t\tWaveformLength: "<<fedRiderCH.WaveformLength()<<std::endl;
    std::cout<<"\t\tTrigNum: "<<fedRiderCH.TrigNum()<<std::endl;
    std::cout<<"\t\tDataIntCheck: "<<fedRiderCH.DataIntCheck()<<std::endl;
}
```

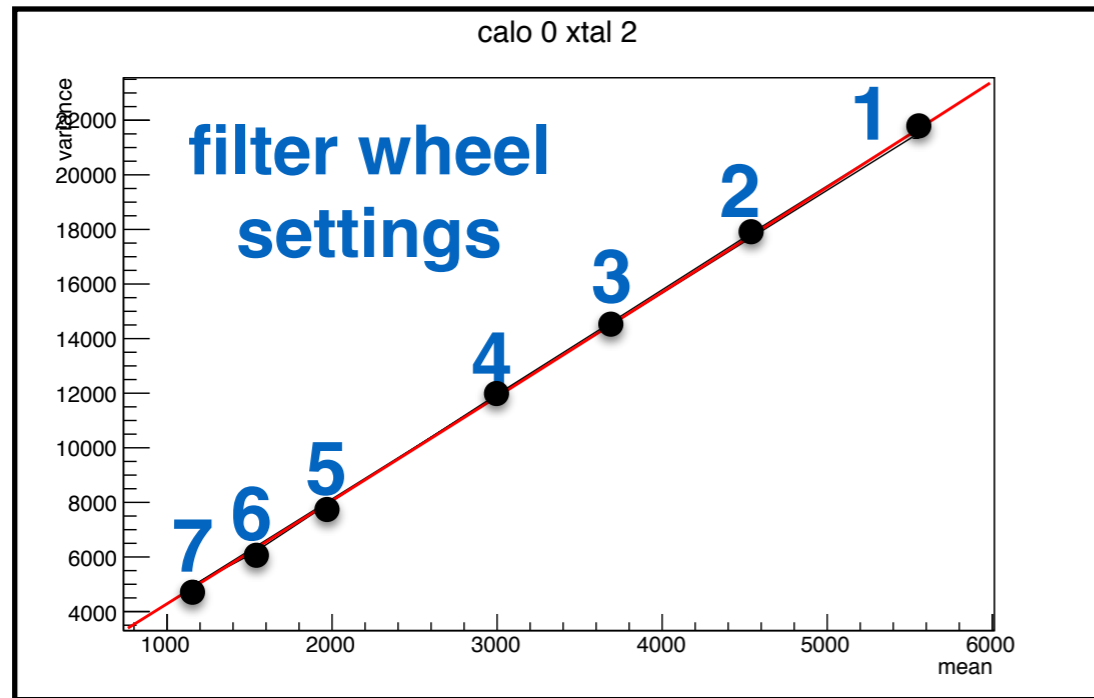
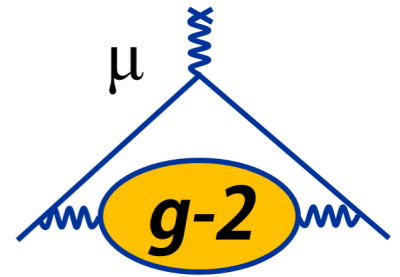
Fill event topology



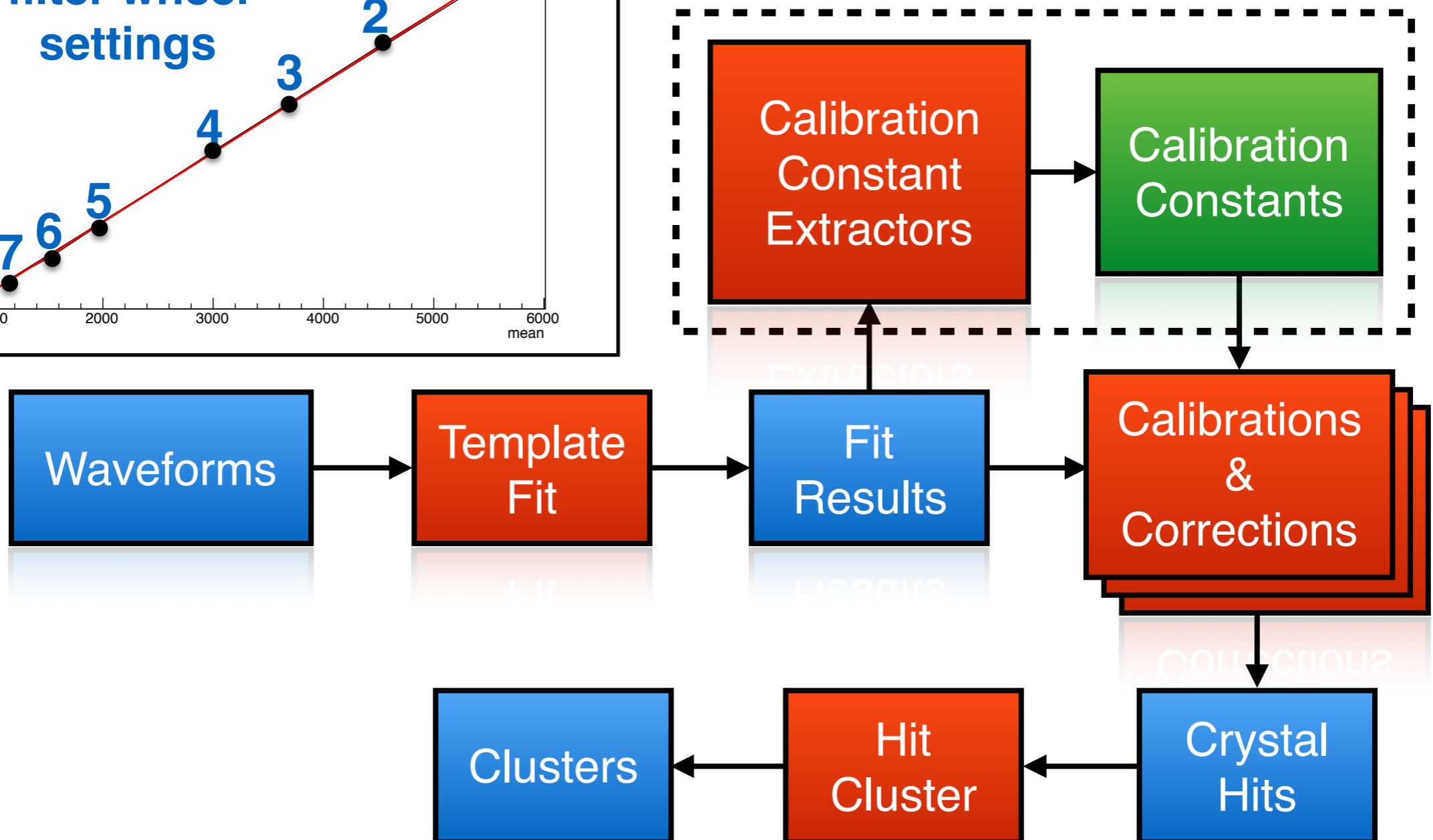
Data reconstruction



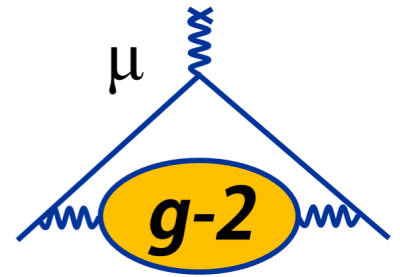
Energy calibration



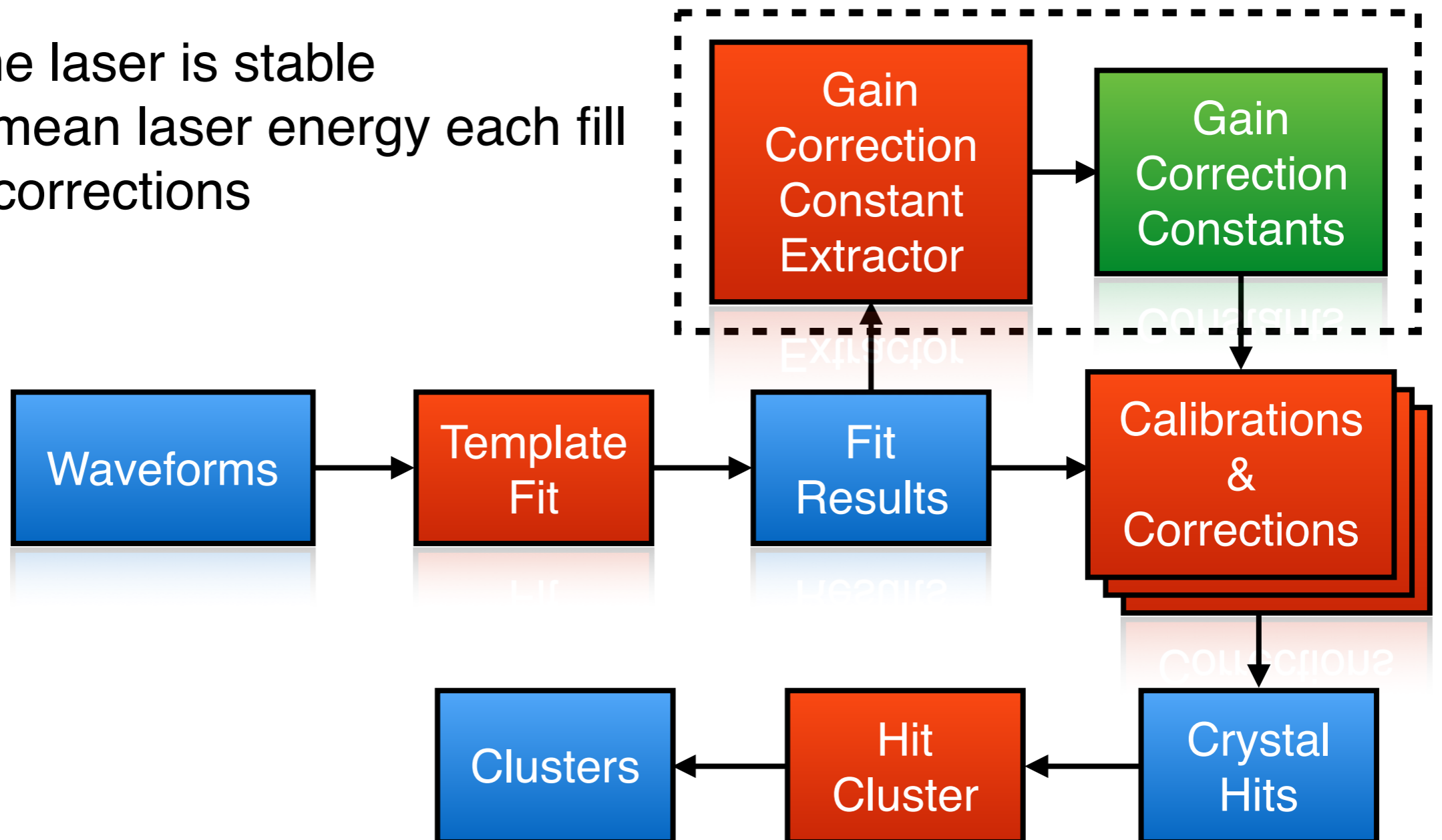
Developed in python,
now incorporated into art.
Constants stored in fhicl prologs.



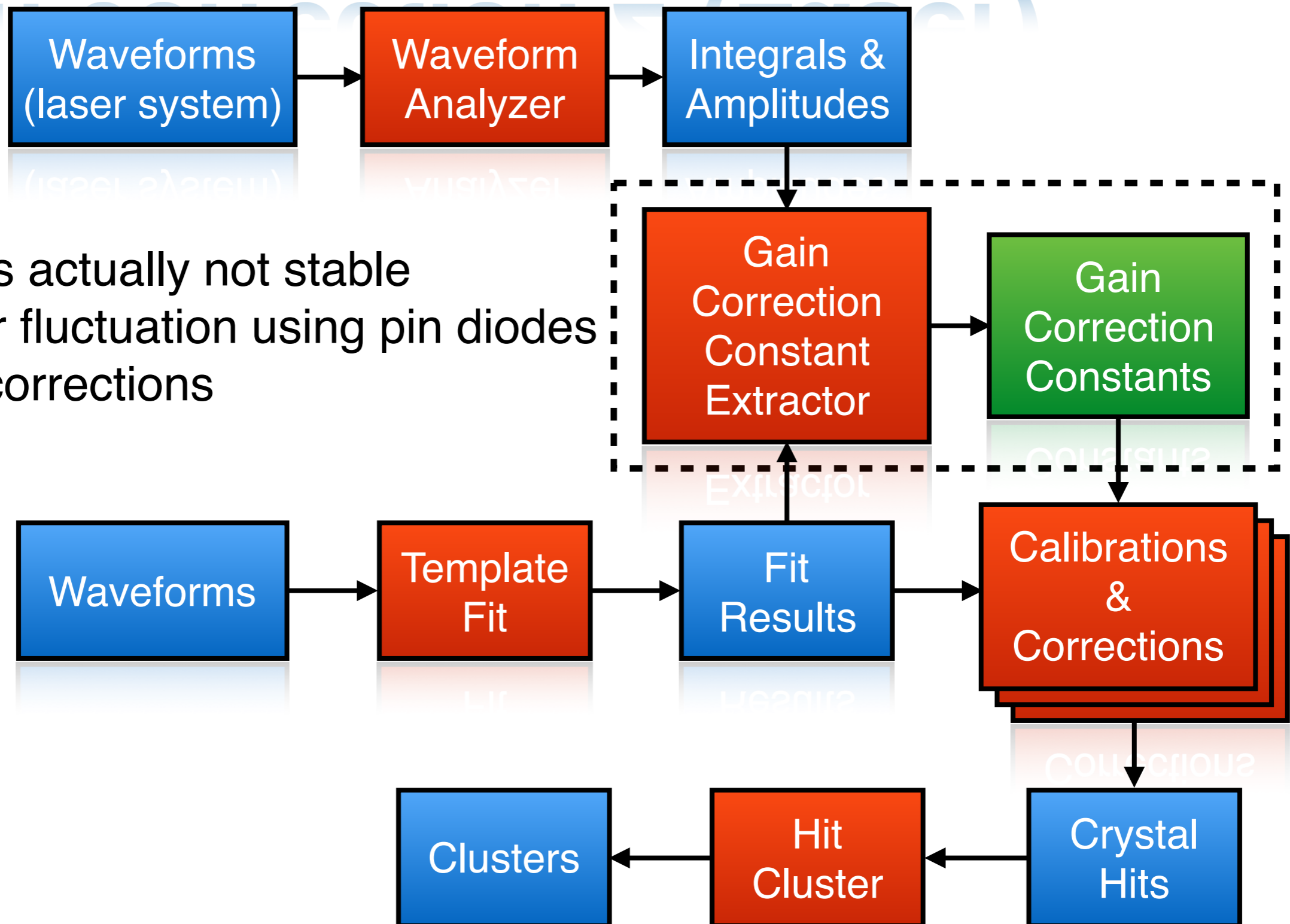
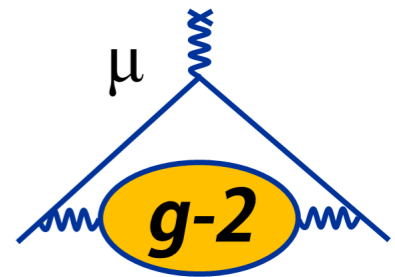
Gain correction 1 (SiPM)



- Assume laser is stable
- Track mean laser energy each fill
- Apply corrections

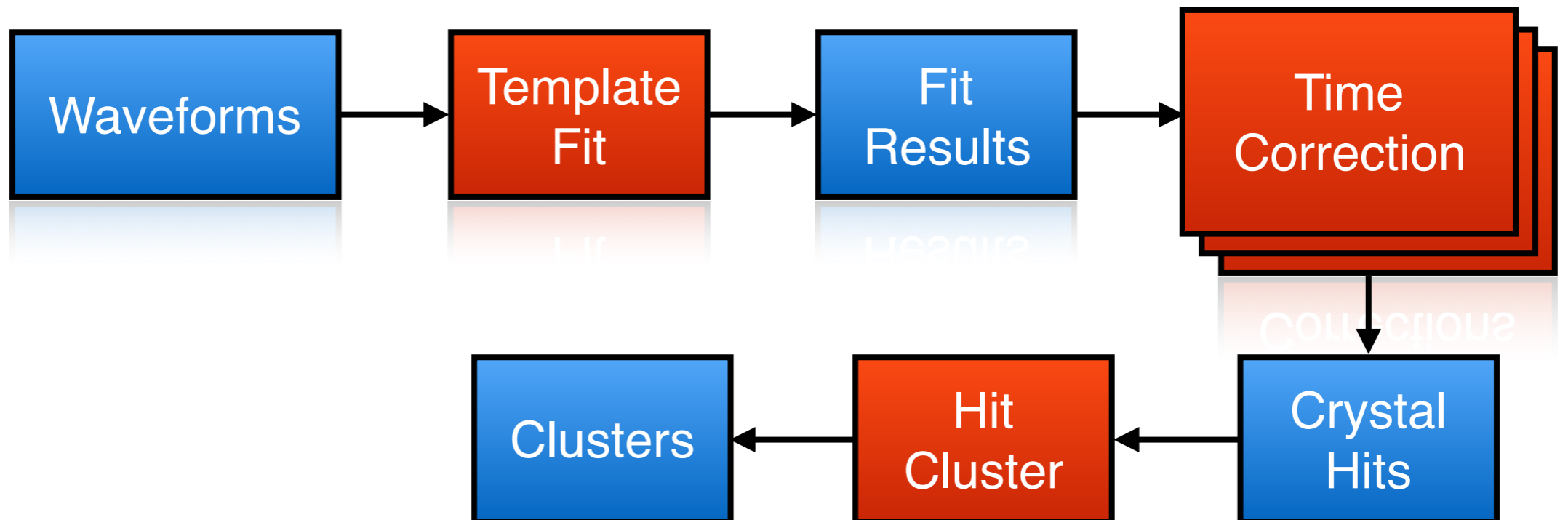
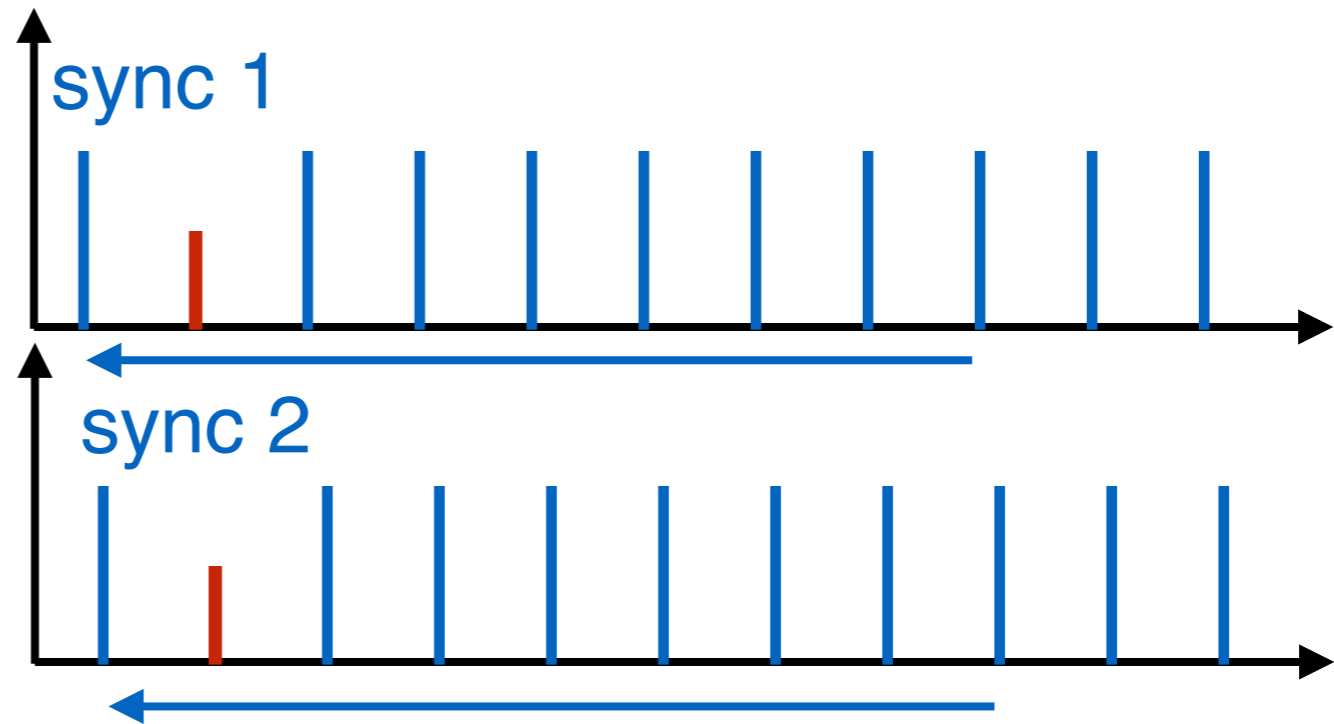
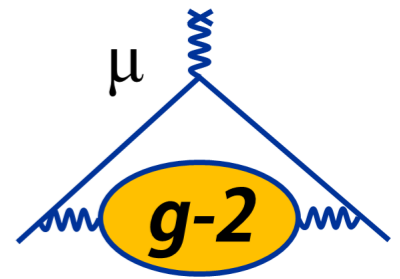


Gain correction 2 (Laser)

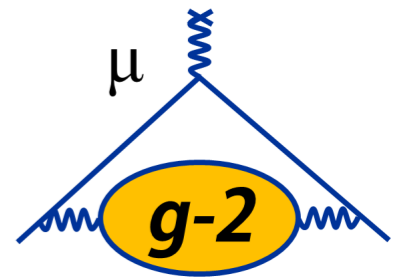


Time correction

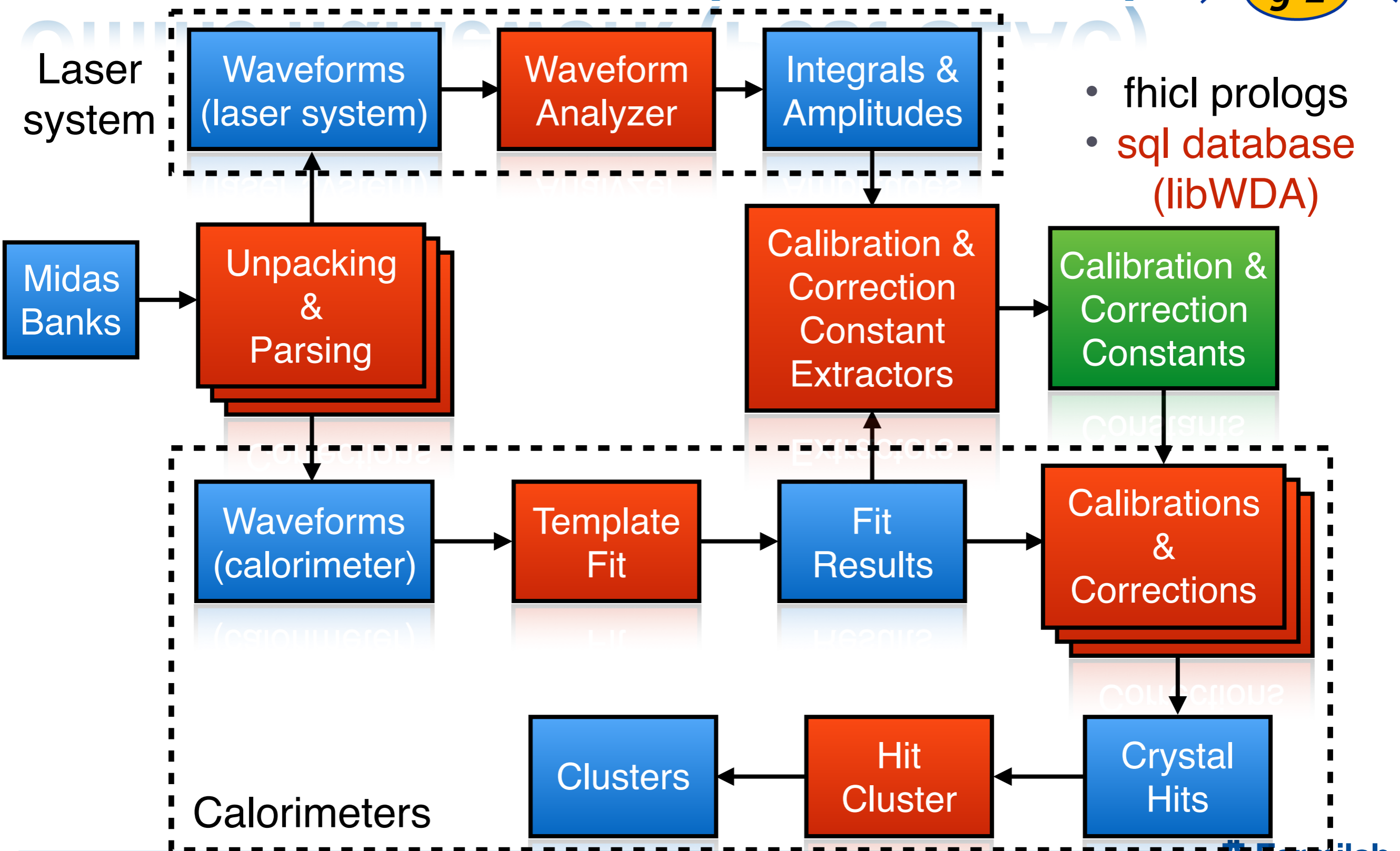
- The digitized waveforms are not time aligned
- Laser sync pulses are configured to arrive after the “begin-of-fill”
- Align time in each channel to this pulse on per-fill basis



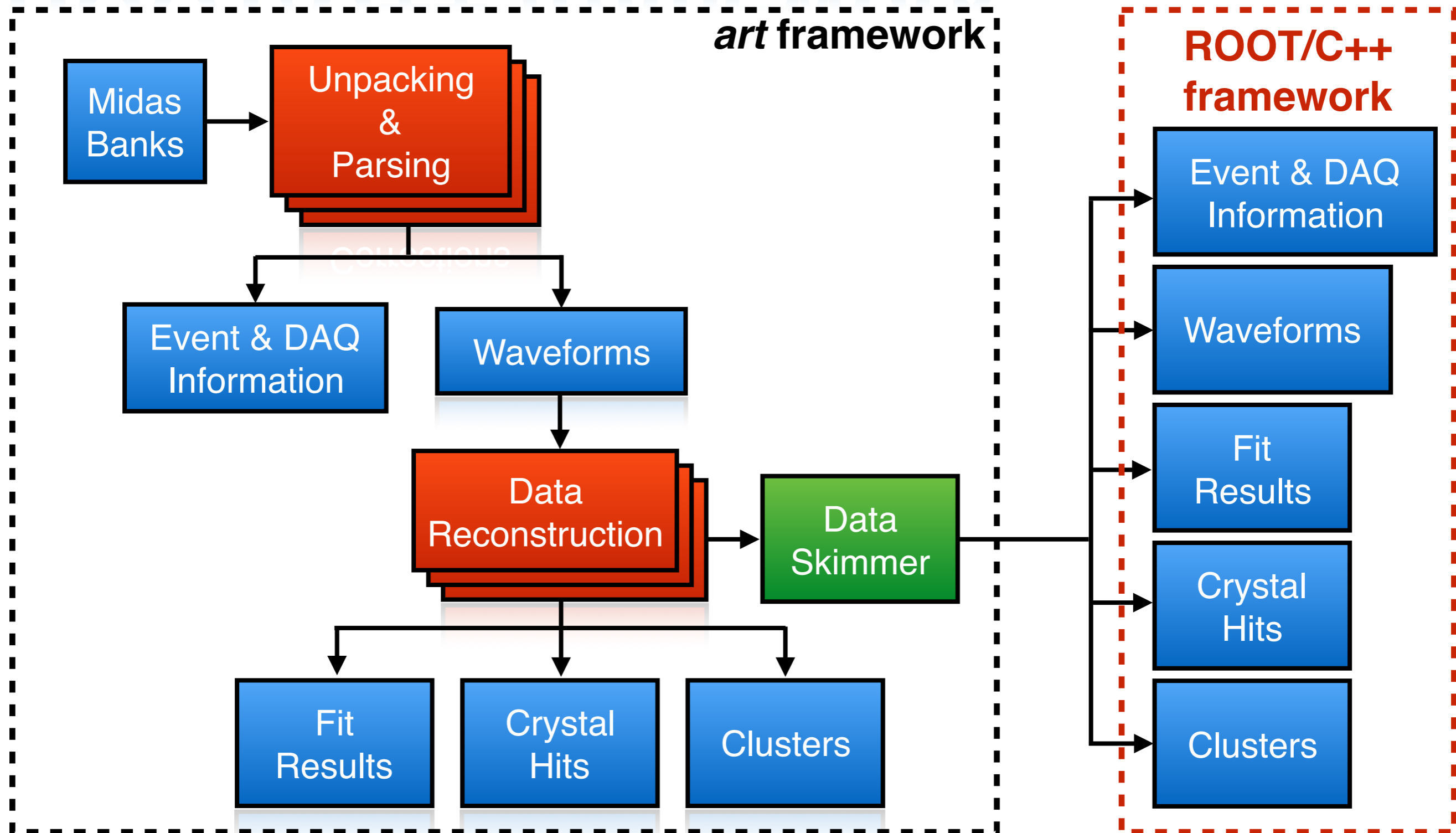
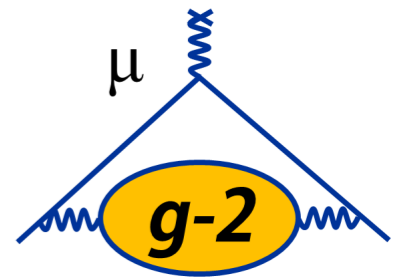
Offline framework (Post-SLAC)



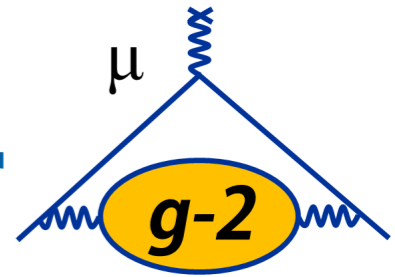
- fhicl prologs
- **sql database** (libWDA)



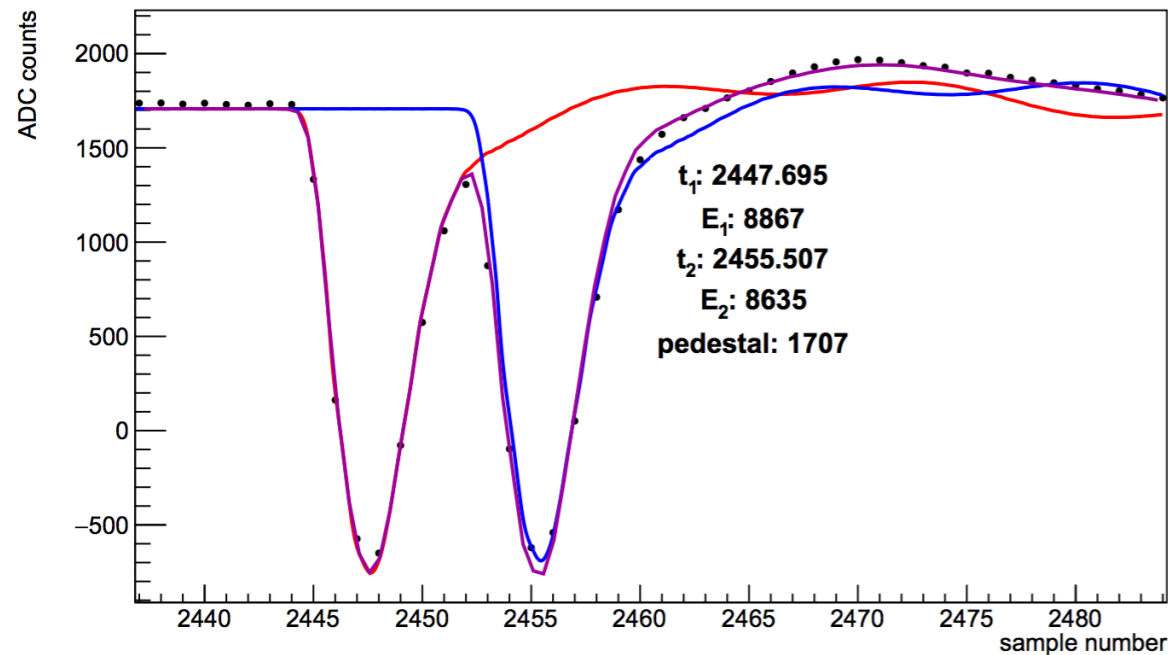
ROOT/C++ framework



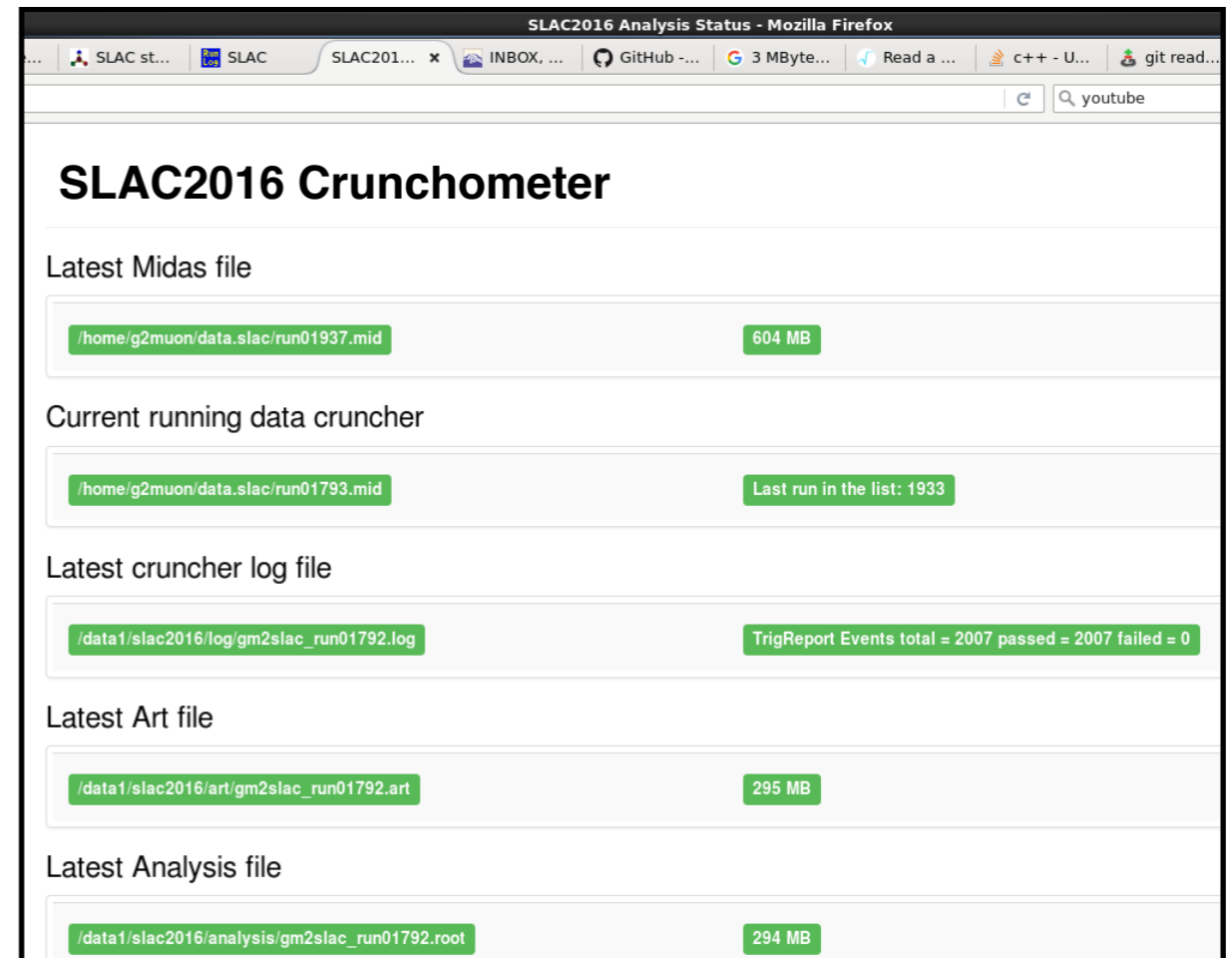
Event display & Data processing monitor



event 17 calo 0 xtal 26 island 9

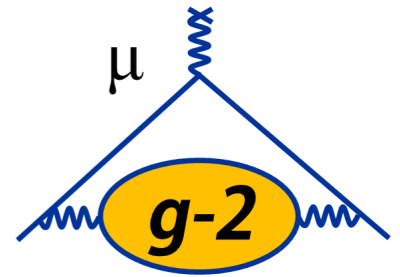


- Simple C++ program using ROOT library
- take root file as input, can plot the waveform and the fit results at the same time
- extremely helpful for DAQ/detector debugging



- Monitoring current MIDAS DAQ file
- Monitoring current file being processed by offline machine

Analysis User Guide

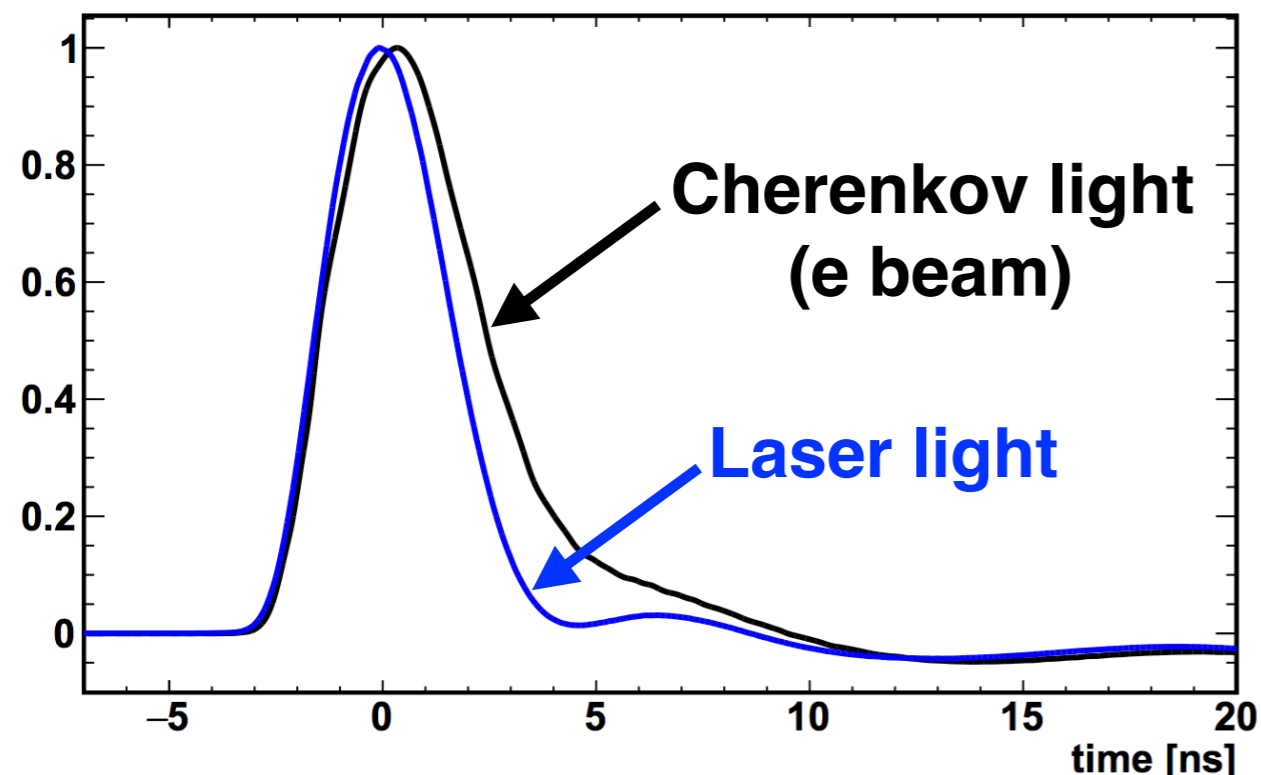
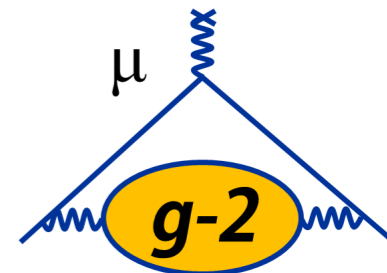


- Similar to Muon $g-2$ computing manual but focus on physics analysis and SLAC specific info
- Also documented the measurement program to aid the analysis
- A good starting point for future analysis guides

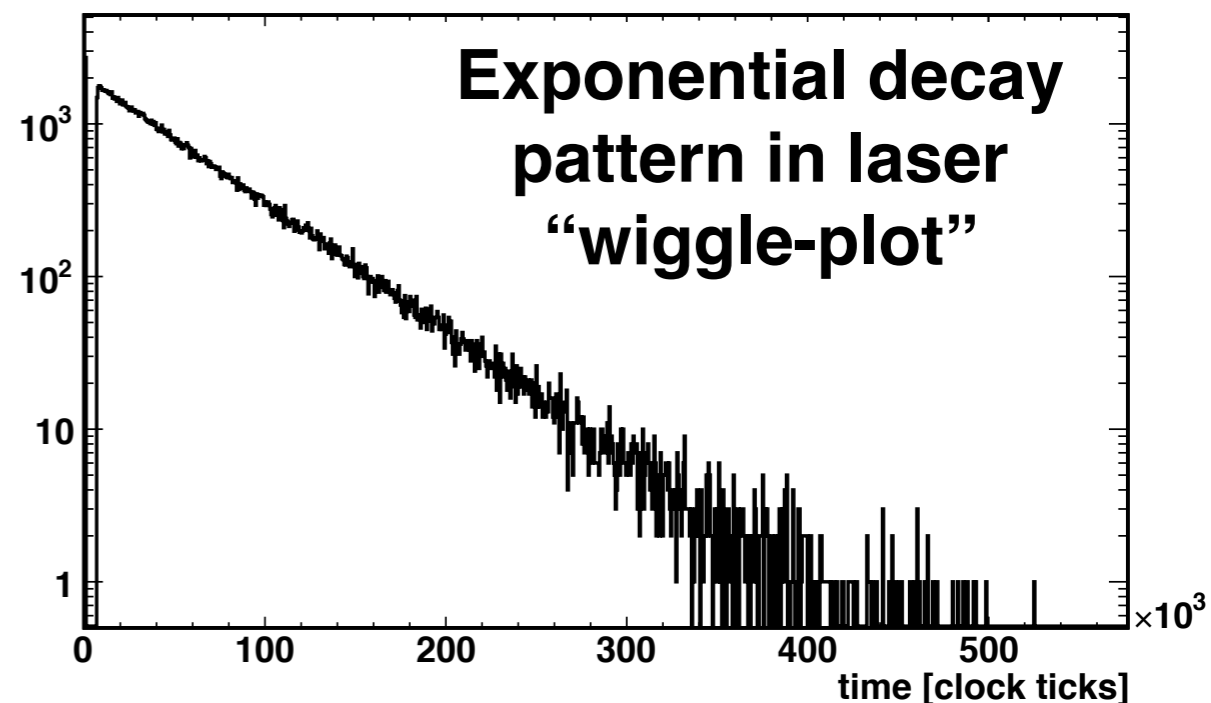
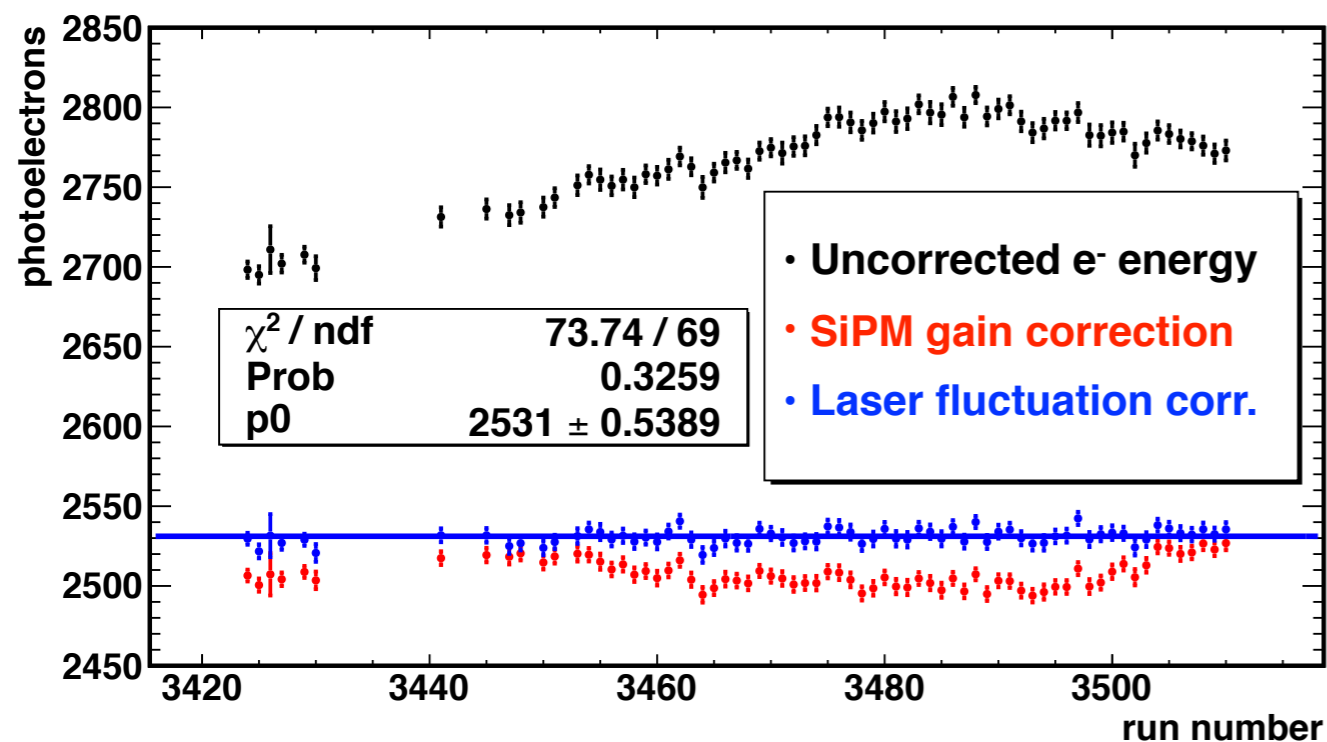
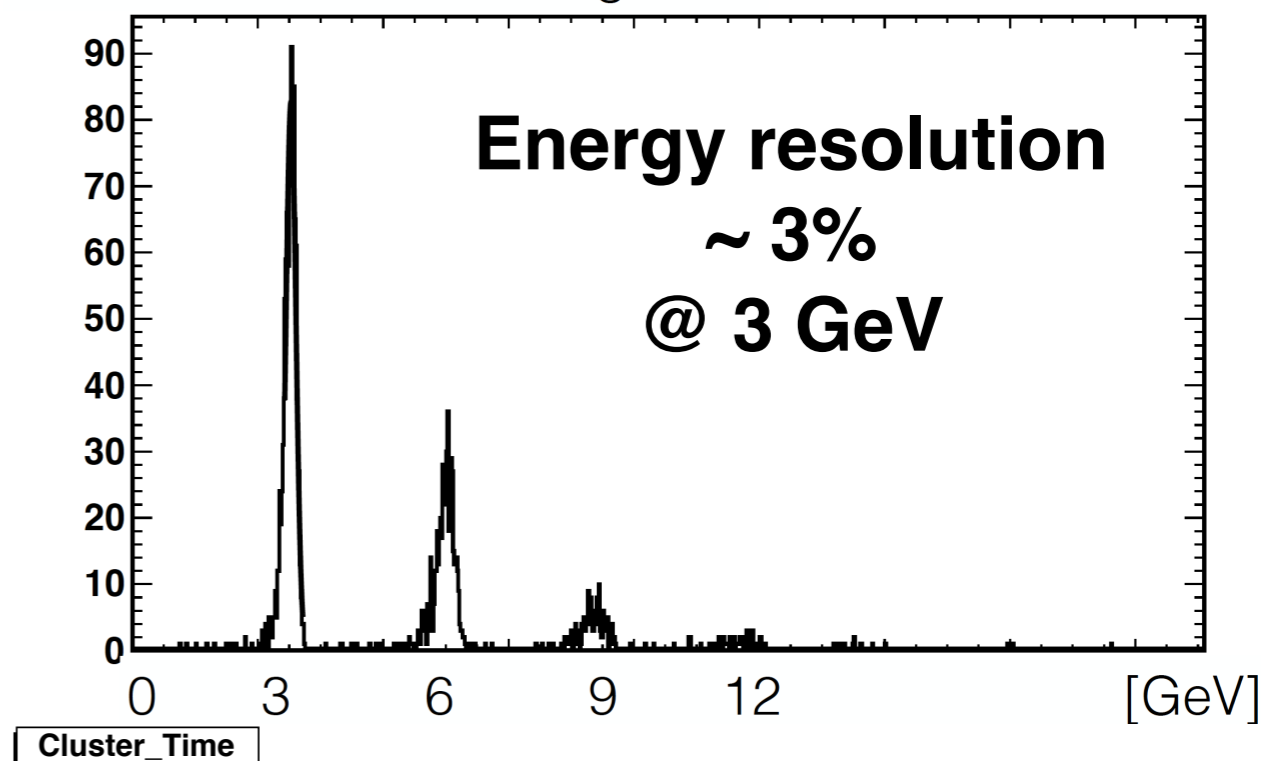
<https://github.com/kimsiang/SLAC2016/tree/master/AnalysisManual>

KIM SIANG KHAW, UNIVERSITY OF WASHINGTON
JASON HEMPSTEAD, UNIVERSITY OF WASHINGTON
THOMAS STUTTARD, UNIVERSITY COLLEGE LONDON

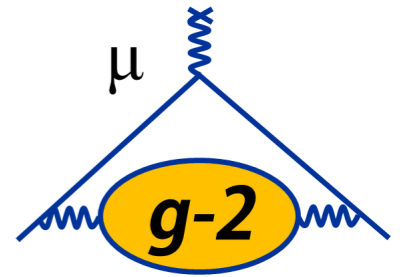
Preliminary results



Poisson comb of hit energies

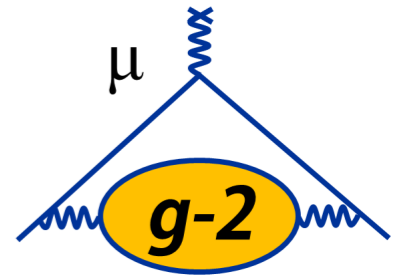


Production

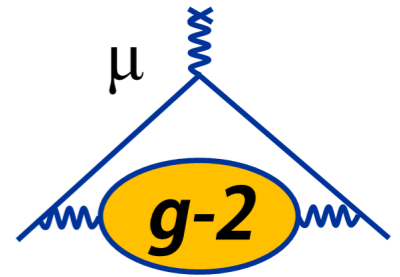


- Dataset collected at SLAC is great for practicing the production routine
- We used Fermi FTS to transfer the raw MIDAS file to dCache tape-backed area
- Since the offline processing rate was close to the DAQ rate (12 Hz), 4 weeks of test beam \sim 4 weeks of processing on single-core
- Initially we have used only jobsub to reprocess all the files
- Then we used jobsub + SAM + ifdh to reprocess a subset of the files (tested all the Fermilab supported services we will be using for the experiment)
- All the reprocessed files are stored in the dCache persistent area

Summary

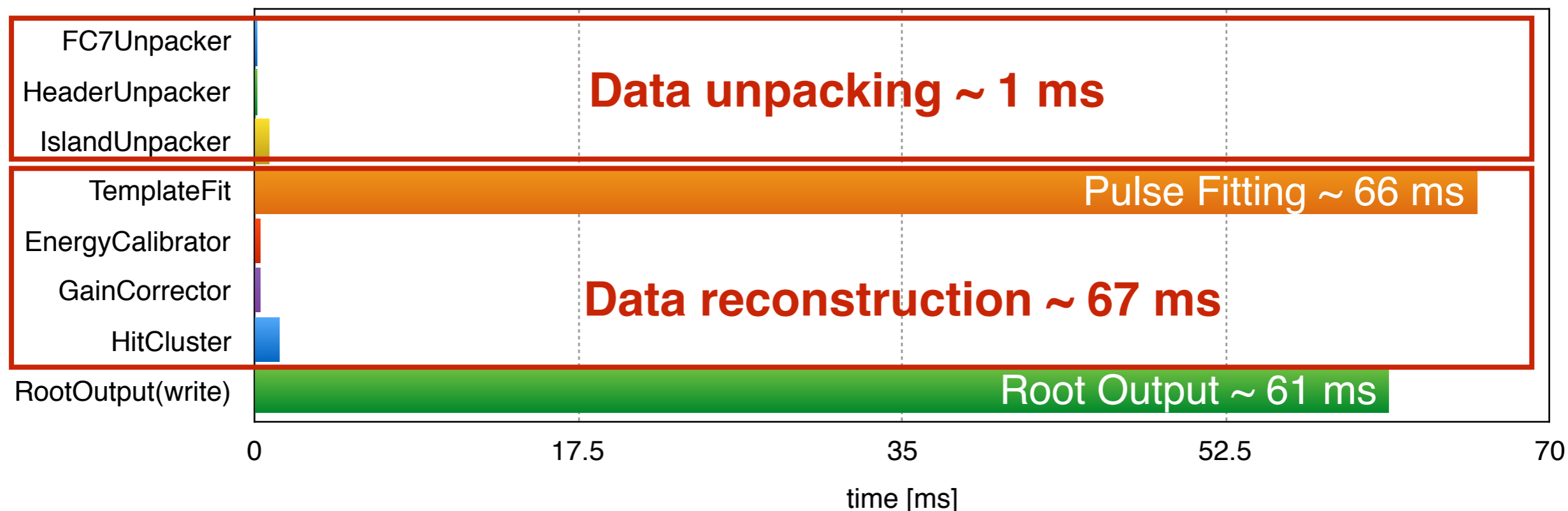


- We had a successful end-to-end test run of the calorimetry system at SLAC in summer 2016
- The offline framework worked reasonably well and provided many valuable feedbacks to the calorimetry system
- We have developed several user friendly tools to aid the data analysis (ROOT/C++ framework, event display, data processing monitor, analysis user guide)
- We have tested the full production chain (FTS, jobsub, SAM, ifdh)
- Analysis is still ongoing, results expected ~ 3 months from now
- Based on our experience at SLAC, we are working towards a fast turnaround physics analysis framework (nearline analysis) utilizing TBB multithreading



Backup

Timing test (as of this review)



TimeTracker printout (sec)	Min	Avg	Max	Median	RMS	nEvts
Full event	0.0606531	0.0689881	0.501452	0.063798	0.0435458	100
unpackerPath:fc7Unpacker:FC7Unpacker	7.604e-05	0.000102685	0.00109262	9.0768e-05	0.000100388	100
unpackerPath:headerUnpacker:HeaderUnpacker	8.3824e-05	0.000107083	0.000941674	8.80715e-05	8.62767e-05	100
unpackerPath:islandUnpacker:OnlineIslandUnpacker	0.000642255	0.000724877	0.00135163	0.000711773	7.16665e-05	100
unpackerPath:islandFitter:TemplateFit	0.0579027	0.0661387	0.494832	0.0609873	0.0431672	100
unpackerPath:energyCalibrator:EnergyCalibrator	0.000242105	0.000267431	0.000674476	0.000253493	4.61028e-05	100
unpackerPath:gainCorrector:GainCorrector	0.000232165	0.000259864	0.000460291	0.000242228	3.48295e-05	100
unpackerPath:hitCluster:HitCluster	0.00115472	0.00130267	0.00187135	0.00128985	8.2678e-05	100
unpackerPath:TriggerResults:TriggerResultInsertter	1.3608e-05	1.48846e-05	3.0534e-05	1.41715e-05	2.91178e-06	100
end_path:outfile:RootOutput	2.015e-06	2.57126e-06	1.4517e-05	2.267e-06	1.51856e-06	100
end_path:outfile:RootOutput(write)	0.0597246	0.0613416	0.0658187	0.0611854	0.0011628	100

Current processing time (single-core)

* SSD/HDD = 0.7

Conditions	Offline processing [ms]	*RootOutput [ms]
1 calorimeter	69	42/61
24 calorimeters	1656	1008/1464
50% less pulses (Muon g-2)	828	504/732
Total	1332/1560 ms	

- 0.75/0.6 Hz (if we write out everything)
- 1.2 Hz (if we skip the root file output)
- write speed ~ 10 MB/s (SSD, c.f. gm2 simulation ~ 40 MB/s)

Q: How do we go from 1 Hz towards 12 Hz?